■ La storia di Linux ■ Giocare a dadi con la Raspberry Pi

SETTEMBRE 2015

LINUX PRO

Diventa protagonista della programmazione Open Source

Raspberry Pi
Crea una stazione meteo!

WINDOWS 10 HA COPIATO!

Ma Linux è gratis e funziona molto, molto meglio!



DVD ROM 8GB

SCOPRI SUBITO IL SISTEMA OPERATIVO FACILE DA USARE E BELLO DA VEDERE!

Korora 22

I migliori programmi di chat
Ecco cosa usare per comunicare con gli amici al meglio **p.54**

Salvare il mondo!
Ecco come un robot Open Source preserva la storia dell'umanità **p.20**

ACCADEMIA DEL CODICE
8 pagine di pura programmazione da pagina 80

Linux Pro 157 - 2015 - Mensile - 5.90 €
9 771722 616763 50157
TARIFFA R.O.C. - POSTE ITALIANE SPA - ABB. POST. - D.L. 353/2003 (CONV. IN L. 27/02/2004 N. 46) ART. 1, COMMA 1, S/NA - DISTRIBUTORE: PRESS-DI, MILANO
Sprea Editori

Quando il cacciatore diventa la preda...

Linux nasce come sistema fondamentalmente basato su linea di comando. Non solo storicamente, ma anche filosoficamente Linux ha sempre basato la sua forza anche sulla snellezza e sulla rinuncia a inutili orpelli. La linea di comando però non è per tutti gli utenti di livello base e anche per questo motivo sistemi come Windows e Mac OS riuscirono a catturare un vasto pubblico. Nel corso degli anni diverse distro Linux hanno sempre più lavorato a perfezionare l'interfaccia grafica per conquistare chi voleva un sistema più semplice da usare e chi ne voleva uno che gli ricordasse i sistemi commerciali. In questo senso Linux ha spesso inseguito le sue controparti a pagamento per offrire una semplicità d'uso e un impatto grafico analogo. Windows insomma era la preda e Linux il cacciatore. Quest'anno è successo però qualcosa di storico, specificamente l'uscita di Windows 10. Però è un fatto storico anche per il mondo Linux? Perché il nuovo sistema Microsoft si vanta di aver portato ai suoi utenti una vasta serie di novità che sono pensate per migliorare la loro vita informatica. "Peccato" che molte di quelle novità siano in realtà nate nel mondo Linux anni fa! Come vediamo nell'articolo di copertina di questo numero, Microsoft ha decisamente preso ispirazione dal mondo Open Source per ai suoi utenti funzioni e programmi prima appannaggio di Linux. Il messaggio per noi è chiaro: la preda è diventato il cacciatore! Sì, per anni Linux ha cercato di aggiungere funzioni e di semplificare il proprio approcciare per abbracciare il vasto mondo degli utenti che pagavano per il loro sistema operativo. Oggi i ruoli sono ribaltati: è Windows che cerca di aggiungere funzioni nate nel mondo Open Source per recuperare popolarità. Se l'imitazione è la più sincera delle adulazioni allora non possiamo fare altro che essere felici nel vedere che è Linux a dettare i ritmi del futuro dei sistema operativi per tutti gli utenti!

La redazione di Linux Pro





CONTATTI

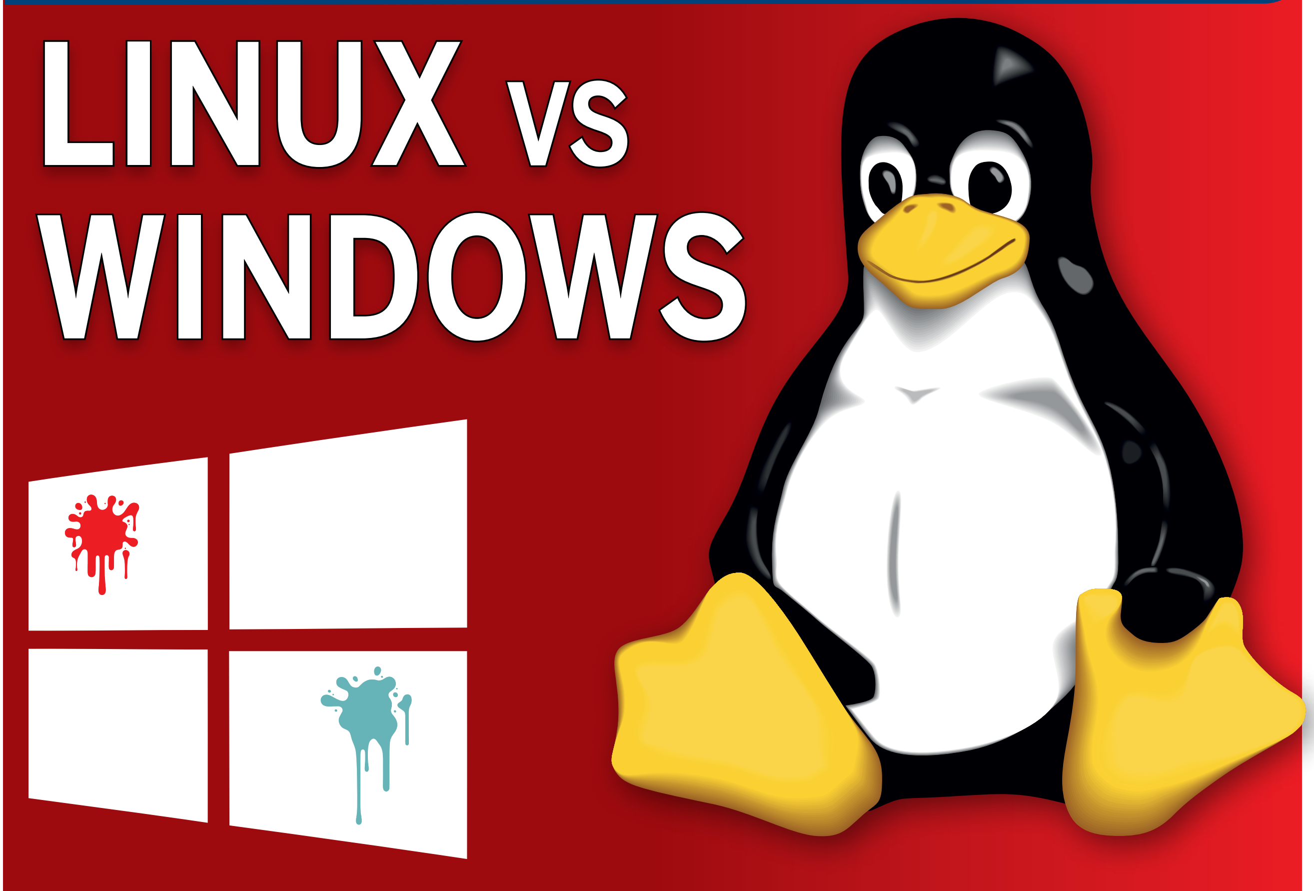
Domande alla redazione: **redazione@linuxpro.it**
Abbonamenti: **abbonamenti@linuxpro.it**
Arretrati: **arretrati@linuxpro.it**
Problemi con il DVD: **aiutocd@sprea.it**
Sito Web: **www.linuxpro.it**
Oppure inviate le vostre lettere a:
***Linux Pro*, Sprea S.p.A.,**
Via Torino 51, 20063 Cernusco S/N
Telefono: 02.92432.1

LINUX PRO 157

Sommario

Benvenuti nel centocinquantasettesimo numero di Linux Pro, la guida definitiva a Linux e al mondo Open Source

In primo piano





Sommario

IL PROSSIMO NUMERO ESCE IL 13 OTTOBRE

IL DVD IN BREVE

LATO A

DISTRIBUZIONI
- Korora 22

LATO B

DISTRIBUZIONI
- REMnux

RIVISTA
- Audacity 2.1.0
- Bracket 1.2
- Chromixium 1.5
- Empathy 3.12.3
- Fiziko 0.1
- FreeCol 0.11.5
- Freemind 1.0.1
- Geary 0.10
- Jitsi 2.9
- Kopete 0.12.2
- Linphone 3.8.5
- Netsurf 3.3
- Pasang Emas 5.0
- Pidgin 2.10.11
- Psi 0.15
- Shotwell 0.22.0
- VirtualBox 5.0 Beta
- VWarsow 1.51

Quando trovi questo bollo negli articoli, cerca il software nel DVD

LINUX PRO dentro DVD

Newsdesk

Ogni mese tutte le novità dal mondo delle aziende e della comunità Open Source

Se hai news da segnalarci o dei commenti scrivi a newsdesk@linuxpro.it

KDE

Android su Linux

KDE sta lavorando per convertire le app nell'ambito del progetto Shashlik

Sebbene Android sia un sistema operativo basato su Linux, le differenze con i suoi cugini Ubuntu, Fedora, ecc. sono piuttosto marcate. La principale è sicuramente l'incompatibilità tra le applicazioni. Infatti, pur essendo figli dello stesso kernel, finora è stato necessario usare strumenti come **ARC Welder** per permettere alle app di Android di funzionare, per esempio, in Ubuntu. Tra l'altro, questo strumento funziona esclusivamente con Google Chrome, con tutti i limiti che ciò comporta. Il problema sta tutto nel fatto che, mentre i sistemi operativi Linux che usiamo sono basati sulle librerie GNU, quello di Google ha le proprie. Lo scorso 26 luglio è stato però presentato un nuovo progetto Open Source che, almeno nelle intenzioni, dovrebbe gettare un ponte tra Android e gli altri sistemi operativi basati su Linux. Si tratta di **Shashlik**, che viene sviluppato dalla community di KDE. L'obiettivo è quello di poter avviare nei nostri computer le app del sistema operativo mobile di Google, come se fossero native. In pratica, si tratta di un emulatore Android e, per il momento, le esperienze precedenti sono solo parzialmente incoraggianti, visto che non è stata mai raggiunta la compatibilità totale. Quindi, la nostra speranza è che la community di KDE riesca laddove altri si sono dovuti arrendere. Tuttavia è presto per trarre conclusioni, poiché il progetto è ancora giovane e deve farsi le ossa. Noi ovviamente facciamo il tifo per lui. LXP



La community di KDE sta lavorando a Shashlik, un progetto Open Source che vuole rendere compatibili con Linux le app di Android

Un grande kernel Linux

Linux 4.2-rc1 è finalmente disponibile e, considerata la gran quantità di nuovo codice inserito, questo kernel risulta piuttosto voluminoso. Secondo Linus Torvalds, se si contano solo le linee aggiunte, che sono oltre il milione, si tratta della release candidate più grande mai vista. Uno dei motivi di queste dimensioni è dovuto alla presenza dei nuovi driver per la GPU AMD. Inoltre, questo kernel include, in una forma più pulita, l'architettura Renesas H8/300. Comunque, in questa release candidate, sono anche state cancellate circa 250.000 righe di vecchio codice. Inoltre, Torvalds ha commentato che, stranamente, ci sono state un sacco di modifiche di basso livello per l'x86, cosa abbastanza inusuale, poiché quel tipo di codice è di solito molto stabile e non necessita di grandi cambiamenti. Per avere maggiori informazioni sulle novità riguardanti Linux 4.2-rc1, colleghiamoci all'indirizzo **https://lkml.org/lkml/2015/7/5/218**.



Con un milione di righe in più, la release candidate di Linux 4.2 è la più voluminosa mai realizzata finora

Mobile

Android sotto i 40 euro

Arrivano i dispositivi economici di Google: avranno successo?

Gli **Android One** sono gli smartphone economici lanciati da Google nell'autunno del 2014 per cercare di conquistare il mercato asiatico e soprattutto quello indiano, che oggi può vantare quasi 500 milioni di potenziali utenti. L'idea di Google era quella di occuparsi direttamente di design e progettazione, lasciando la costruzione dell'hardware a produttori locali. In realtà i margini per i produttori erano talmente bassi che i pochi modelli arrivati sul mercato si sono rilevati di qualità inferiore anche rispetto alle esigenze di un'utenza bassa. Ora, a distanza di un anno e con l'inevitabile discesa dei prezzi delle componenti, Google ci riprova cercando di rimanere sotto la soglia dei 40 euro, usando componenti di buon livello come il **SOC** di Mediatek con processore quadcore e display in formato 720 p, risparmiando semmai sulla quantità di memoria disponibile. Il tutto al momento rimane destinato agli utenti asiatici, ma se il prodotto avrà successo non è detto che non possa sbarcare anche in Europa. LXP



› Dopo una prima versione, che nonostante il prezzo ridotto al minimo ha avuto poco successo, Google ci riprova dotando *Android One* di componenti di migliore qualità e mantenendo il prezzo sotto la soglia dei 40 euro

Unity

Un game editor per Linux

Finalmente gli utenti Linux potranno creare videogiochi con Unity

Unity, il motore grafico alla base di alcuni tra i giochi più noti al mondo, sta per arricchirsi di un game editor funzionante con Linux, che finora era disponibile solo per Windows o Mac OS X. Na'Tosha Bard, uno sviluppatore Linux, nel blog di Unity ha scritto che la creazione di questo editor è stata un "atto d'amore" e ha aggiunto che i risultati ottenuti dal team, che ha partecipato a tempo perso a questo progetto, sono piuttosto buoni. Presto verrà rilasciata una versione sperimentale, mentre quella definitiva potrebbe essere supportata da Ubuntu, che è la distro di Linux più popolare. Per scoprire di più su Unity 5 invece abbiamo la nostra recensione completa a pagina 48. LXP

LINUX VS WINDOWS



Abbiamo provato Windows 10 e lo abbiamo messo a confronto con il mondo Linux. Come è andata?

Lo scontro è epico e oramai il suo inizio si perde negli albori del tempo. Windows contro Linux è una diatriba che infiamma i cuori e che in ciascuno schieramento ha i suoi sfegatati sostenitori. In realtà, le differenze tra i due sono abissali e in larga parte neppure comparabili tra loro. Linux è progettato su base libera e può girare su qualsiasi hardware. Windows è un prodotto proprietario che fino a poco tempo fa poteva essere eseguito solo su PC x86. Dal 29 luglio è però arrivato Windows 10 e non potevamo farci mancare un confronto serrato tra l'ultimo nato in casa Redmond e il Pinguino. Il nostro approccio per la stesura di questo articolo è stato semplice: abbiamo preso una build di anteprima di Windows 10 e l'abbiamo confrontata con alcune distro tra quelle che a nostro parere possono essere definite equivalenti. Microsoft, per il lancio del suo nuovo sistema operativo, ha puntato tutto sui cambiamenti rispetto al passato. Di novità, non possiamo negarlo, ce ne sono, a cominciare dal cambio di filosofia nel considerare Windows un servizio anziché un sistema operativo. Anche sul fronte degli aggiornamenti ci sono delle importanti variazioni. Per gli utenti non verranno rilasciati fino a quando Microsoft non lo riterrà opportuno. Per le imprese, invece, è stata pensata una suddivisione in *Current* e *Long Term* che prevede la diffusione degli update in modo più controllato. Chi acquista Windows 10, o ne verrà in possesso tramite le offerte di aggiornamento gratuito, noterà che il supporto sarà esteso fino alla durata naturale del dispositivo in cui lo si installa.

> **"Abbiamo provato Windows 10 e messo a confronto con distribuzioni equivalenti"**

Un addio al passato?

Windows Vista è stato uno dei sistemi più contestati della storia di Microsoft. Sotto questo aspetto può forse essere paragonato al flop di Windows 8. Al contrario di quest'ultimo, però, aveva requisiti minimi hardware eccessivamente alti e in più introduceva in modo sbagliato il concetto di Controllo dell'Account Utente (UAC). Le continue interruzioni a cui eravamo sottoposti per fornire privilegi amministrativi avrebbero fatto perdere la pazienza anche a un santo. Tuttavia il concetto alla base del UAC era giusto, così come erano innegabili i vantaggi in termini di gestione grafica che si sono avuti con il supporto per le DirectX 10. A parte questo, però, Vista è stato percepito come un fallimento e quindi ignorato da imprese e utenti. Alla fine, nel momento migliore della sua distribuzione, è riuscito a totalizzare solo un misero 21% di utilizzo.

Windows 8 è stato un altro flop. L'interfaccia grafica pensata per i dispositivi mobile è stata progettata ignorando quasi del tutto chi usava ancora mouse e tastiera. Alcune funzioni, tra cui il pulsante **Start**, sono state prima eliminate e poi ripristinate. Senza poi contare il fallimento di **Windows Store**, rimasto un guscio vuoto o al limite pieno di applicazioni spazzatura. Valutando questi fallimenti, Microsoft ha quindi deciso di puntare tutto su un aggiornamento totale delle proprie piattaforme a **Windows 10**. Vista e XP hanno terminato il supporto e quindi dovranno essere sostituiti. 7 e 8 potranno invece essere aggiornati gratuitamente. Fare piazza pulita delle vecchie versioni di Windows è quindi uno dei principali obiettivi della versione 10. L'uso di una sola release, inoltre, permette a Microsoft di concentrare i propri sforzi su aggiornamenti mirati e non più frammentati. In più aiuta gli utenti a fare i conti con un solo ambiente. A livello estetico, Windows 10 non è poi così diverso da Windows 8.1. Sul lato tecnico, al di là delle nuove funzioni presenti e che verranno, troviamo ancora un'infrastruttura vecchio stile. Scavando un po' più nel dettaglio, scoprirete che il gestore periferiche è lo stesso fin dai tempi di XP, mentre i file **win.ini** e **system.ini** sono sempre al loro posto e risalgono ai tempi di Windows 3.1. Certo, l'attuale Microsoft è molto diversa da quella degli anni passati. Seppure sia ancora leader nel campo dei desktop, il suo vantaggio su questo fronte non è più sufficiente per impensierire i concorrenti. Il futuro è mobile e se Microsoft vuole sopravvivere deve imboccare una strada che gli consenta di andare oltre il classico PC da scrivania o il portatile. Infatti, una delle caratteristiche più pubblicizzate di Windows 10 è la convergenza: il sistema può funzionare e mettere in comunicazione PC, smartphone Windows, Xbox, Surface Hub e Raspberry Pi. Tutti questi dispositivi potranno lavorare con un sistema unificato, dove le app saranno disponibili su ciascuna piattaforma senza distinzione di funzioni o caratteristiche. Quando Windows 10 Mobile verrà lanciato, potrete collegarlo a un TV con l'uso di mouse e tastiera, come se fosse un tradizionale PC. Nel 2014, Nadella ha confermato che il 90% delle API poteva sovrapporsi tra mobile, desktop e codice Xbox. La convergenza è anche uno dei cavalli di battaglia di Canonical e sebbene Microsoft riuscirà quasi sicuramente a svilupparla prima e meglio, il vero obiettivo sarà conquistare gli utenti. »

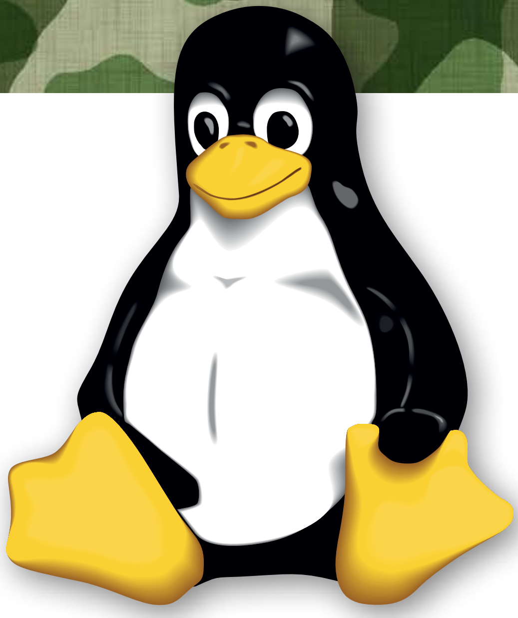


› La GUI di Windows 10 è una via di mezzo tra Windows 7 e Windows 8

Windows, una storia fatta di versioni

Windows 7, rilasciato tre anni dopo Windows Vista, ha fatto un buon lavoro nel tentare di risolvere i tanti problemi creati dal predecessore. Rispetto a XP, gli utenti lo hanno inizialmente percepito con molta diffidenza, ma con il trascorrere del tempo è riuscito a scalare le vette dell'Olimpo. Purtroppo per Microsoft, c'è ancora una grande fetta di utenti che non ha intenzione di smuoversi da Windows XP. Questo nonostante il supporto ufficiale sia oramai cessato. Ancora oggi, dopo 14 anni dal suo rilascio e dopo più di un anno dalla sua *End Of Life* (EOL), XP è ancora una delle piattaforme più utilizzate. Naturalmente, non c'è dubbio sul fatto che Microsoft continui a guadagnarci sopra grazie ai costosi accordi post-EOL, ma questo non toglie che tali risorse potrebbero essere meglio sfruttate altrove. Ciò ci porta al 2012, Windows 8 e alla sua interfaccia **Metro**. Questa versione del sistema operativo Microsoft ha tentato di rivoluzionare l'uso del PC, fallendo poi miseramente. Prendendo in considerazione solo chi usa dispositivi touch, gli utenti che erano ancora legati a mouse e tastiera hanno preferito rimanere a Windows 7. Senza poi contare la rimozione dei punti di riferimento cui le persone erano abituate. Uno per tutti può essere citato il pulsante **Start**, poi fortunatamente ripristinato con Windows 8.1. Quest'ultimo è stato infatti rilasciato appena un anno dopo Windows 8, prendendo in considerazione l'enorme mole di lamentele e cercando quindi di porvi rimedio. Rispetto agli utenti, le aziende sono in condizioni forse peggiori. Molte sono ferme a Windows XP e si scervellavano se passare al 7 o tentare l'azzardo con l'8.1. Nel primo caso, si tratta pur sempre di un sistema operativo con già sei anni sulle spalle, il cui supporto mainstream si è concluso all'inizio del 2015. Certo, fino al 2020 non ci saranno problemi per gli aggiornamenti, ma data la lentezza di alcune organizzazioni, non si tratta poi di un traguardo così lontano. Windows 8.1, almeno fino a ora, ha quindi rappresentato un'alternativa più fattibile nonostante tutti i suoi difetti. Tuttavia l'approccio di Microsoft sotto questo profilo è chiaro: permettere a chi ha l'8.1 e il 7 di tagliare la testa al toro e passare direttamente alla 10. Il fatto poi che le aziende decidano di fare il grande salto in tempi più o meno ragionevoli è un'altra storia.

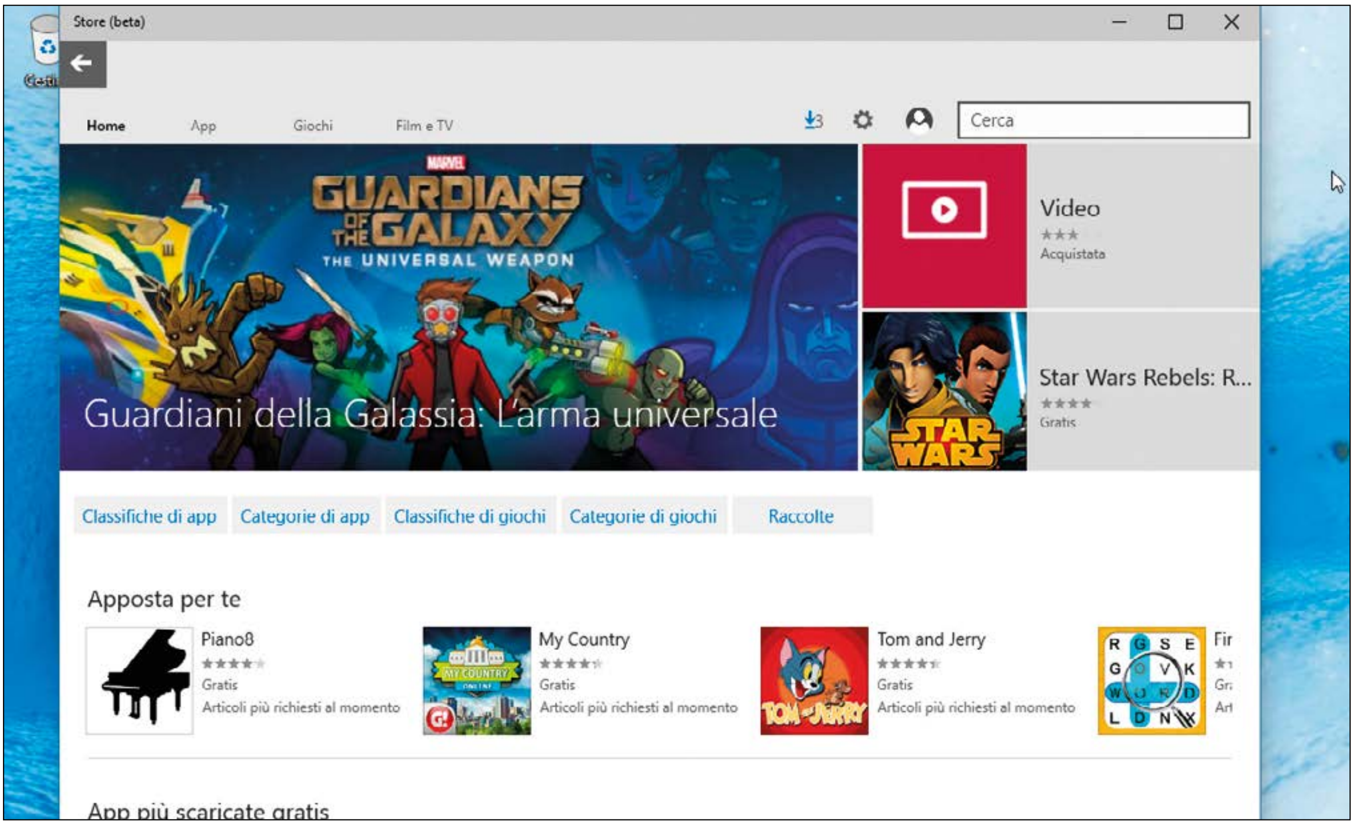
Funzioni familiari



Al giorno d'oggi, le persone usano in modo sempre più forsennato il proprio desktop. Lo riempiono di finestre, applicazioni, widget e quant'altro. Certo, rispetto a quando i display avevano dimensioni contenute, di acqua sotto i ponti ne è passata. Con monitor da 27 pollici, risoluzioni Full HD e adesso Ultra HD le cose sono migliorate, ma spesso non tanto quanto ci sarebbe bisogno. Ecco perché l'idea di usare dei desktop virtuali ha subito preso piede in Linux e adesso anche in Windows.

L'ultimo nato in casa Redmond, per la prima volta dai suoi albori, dispone di una funzione per la generazione dei desktop virtuali. Per impostazione predefinita, abbiamo la scrivania principale con le icone dei vari collegamenti riportati. Tuttavia basta un click per aprire almeno altri tre desktop (in alternativa è possibile usare la combinazione di tasti **Windows+Tab**). Allo stato attuale, questa funzione è però un po' acerba. Usando la scorciatoia da tastiera, infatti, anziché aprire in modo neutro i desktop virtuali, vi trovate a selezionare automaticamente quello principale. Riguardo a questa funzione, va detto che non è una vera novità. I desktop virtuali sono disponibili in Windows fin dai tempi di XP, ma non in modo nativo, bensì attraverso l'uso di applicazioni di terze parti. Il fatto poi che non tutte abbiano funzionato a dovere è un altro argomento. Chi ha esperienza di Linux non potrà fare a meno di essere consapevole che Microsoft ha solo scoperto l'acqua calda. Fin dalle prime versioni di KDE e Gnome risalenti agli anni '90, i desktop virtuali erano già disponibili. Se poi vogliamo risalire ai loro albori, possiamo arrivare fino ai tempi dell'Amiga 1000 (1985) e del windows manager Soldbourne (1990). Il Task View, vale a dire il pannello che riporta i servizi e i programmi operativi, è molto simile a **Gnome Shell Activities Overlay** (la schermata che mostra tutte le applicazioni in esecuzione). Come Gnome Shell, Windows 10 dispone anche di un'area di notifica centrale (definita Centro Operativo). Dalla barra di avvio si è poi in grado di cercare applicazioni e informazioni sul Web, un po' come facciamo con la Dash di Ubuntu. Sotto questo aspetto, essere in grado di

> **"I desktop virtuali vengono usati in Linux fin dagli albori di KDE e Gnome nei lontani anni 90"**



› **Sarà perché è agli inizi, ma rispetto a Ubuntu Software Center, Windows Store è ancora parecchio scarno**

visualizzare tutti i programmi installati è un vantaggio. Questa funzione era vagamente presente anche in Windows 8, ma in modo completamente errato e fuorviante. Niente che possa essere paragonato a quanto Linux faceva già da oltre cinque anni. Windows PowerShell è stato introdotto intorno al 2006. Una delle sue caratteristiche più pubblicizzate consiste nel fornire uno strumento simile al gestore pacchetti.
Il codice sorgente può essere inserito in un pacchetto che all'occorrenza è possibile aprire con un semplice comando.
Il programma che si occupa di questo compito viene chiamato da Microsoft **Software Discovery, Installation and Inventory** (SDII). Allo stato attuale, OneGet è solo un insieme di cmdlet di PowerShell che dialoga con i reposity tramite un'applicazione di terze parti chiamata **Chocolatey Nuget**. Questo fornisce solo 3.000 pacchetti. In futuro, tuttavia, il numero è di sicuro destinato a crescere e molto probabilmente troveremo anche un repo Microsoft. In questo modo, non dovrete più accedere a Internet Explorer per scaricare qualsiasi altro browser. Basterà solo utilizzare una finestra di PhowerShell in

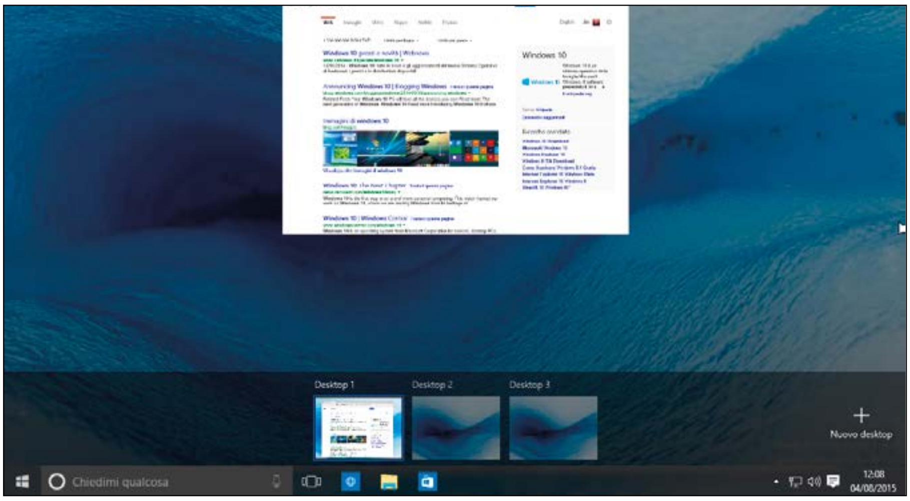


› **Il sistema di Gnome per visualizzare le app è simile a quello di Windows 10**

modalità amministratore e inserire:

```
Install-Package -Name Firefox -Provider chocolatey
```

Per sostituire Firefox con Chrome, abbiamo dovuto argomentare il comando a causa di una disambiguazione con un altro pacchetto chiamato **xFirefox**. Naturalmente, Microsoft incoraggia gli utenti a utilizzare le applicazioni presenti nel proprio App Store, ma i guru della PowerShell potranno comunque divertirsi a gestire la procedura di installazione in modo alternativo. Il concetto di Windows come servizio può in qualche modo essere paragonato a una distro rolling-release in stile Arch Linux o Linux Mint Debian Edition. Allo stesso tempo, il modello del rilascio multi-ramo per le imprese è vagamente simile a quello utilizzato da Debian. Il sistema **Insider Preview**, inoltre, può essere considerato un enorme ambiente di beta-testing, simile a quello che viene utilizzato da Valve per il suo SteamOS. In realtà, tutte queste idee provengono dal mondo Linux e fa molto piacere vedere come un colosso come Microsoft si sia finalmente convinto della bontà dei nostri metodi. Un altro aspetto molto piacevole consiste nella gratuità dell'aggiornamento per chi è in possesso di una licenza valida di Windows 7 o 8/8.1. Questa mossa, tuttavia, va un attimo ridimensionata rispetto alle aspettative dei più. Infatti, si tratta di una politica pensata per attrarre nuove persone e potenziali acquirenti. »



› **Microsoft ha finalmente deciso che i suoi utenti sono maturati abbastanza per affrontare il concetto di Desktop Virtuali. Meglio tardi che mai...**

Windows SSH

Un altro progetto che non fa strettamente parte di Windows è **PowerShell** che tuttavia non dispone ancora di funzioni SSH. Potrete quindi scegliere di collegarvi a Windows tramite la scomoda sintassi di questo strumento, oppure utilizzare un client SSH di terze parti come **PuTTY**. Basterà che sia in esecuzione su un server in cui sia installato l'ambiente Cygwin. Vari estimatori di Microsoft e appassionati di vario genere hanno spinto parecchio perché le funzioni SSH fossero incluse in PowerShell. Il management, tuttavia, non li ha mai voluti accontentare. Nadella è molto più tollerante dei suoi predecessori. Microsoft, infatti, ha intenzione di contribuire alla comunità OpenSSH. La loro collaborazione conta un budget a cinque cifre ed è realmente fattiva. Probabilmente questo comportamento sarà visto con scetticismo dai puristi, soprattutto in virtù dei trascorsi e della filosofia Embrace, Extend, Extinguish. Ciò non toglie che di passi in avanti Redmond ne abbia fatti parecchi negli ultimi anni e anche questo potrebbe essere uno dei tanti.

Prestazioni e grafica

Cerchiamo di essere chiari fin dal principio. La nostra esperienza con Windows 10 è stata costellata da bug e fastidi di vario genere. Si tratta però di una beta e come tale deve essere considerata. A parte questo, il nuovo sistema operativo, una volta installato in un computer ben configurato, si rivela estremamente reattivo.

Ascoltando le tante rimostranze degli utenti di Windows 7, oramai intenti a lamentarsi della lentezza di avvio e spegnimento, qui siamo di fronte a tutta un'altra storia. In Windows 8 è stato introdotto un trucco che permette ai processi di sistema di essere scaricati sul disco in caso di stand-by. In questo modo, nella fase di riavvio, è possibile riattivarli velocemente. In uno stato di sospensione temporanea, solo i processi relativi all'utente devono essere caricati da zero, mentre quelli di sistema sono già disponibili. Ecco perché il tempo di caricamento e ripristino ha raggiunto una velocità encomiabile. La stessa tecnica viene utilizzata in Windows 10 che peraltro è riuscito a eseguire un boot con un SSD in soli sei secondi. Più o meno lo stesso tempo che si impiega per passare da Grub al gestore SDDM di login su una distro Arch. Nonostante l'utilizzo giornaliero piuttosto forzato, Il sistema operativo non ha accennato a rallentare. A questo proposito, però, è necessario fare una precisazione. Noi abbiamo eseguito la prova per una sola settimana. Chi userà Windows 10 per tutta la vita si ritroverà ugualmente pieno zeppo di librerie e aggiornamenti giganti che inevitabilmente porteranno a un calo delle prestazioni. L'unico elemento che ha messo in crisi il sistema, è l'uso massiccio di Flash. Poco male, visto che possiamo disinstallare il plug-in senza ulteriori perdite di tempo. Al giorno d'oggi, infatti, serve veramente a poco, oltre a essere un fastidioso veicolo per virus e affini. Nonostante le specifiche minime hardware sbandierate da Microsoft, per godere al meglio di Windows 10 è necessario un computer abbastanza potente. L'uso su una macchina virtuale è stato a tratti molto doloroso. Sul fronte delle configurazioni più vecchie, non c'è storia. Linux, con i suoi desktop ultraleggeri come LXqt o Mate, vince a mani basse. Basterà poi aggiungere una scheda grafica leggermente più moderna per installare Ubuntu (i requisiti minimi di questa distro sono di 1 GB di RAM e una CPU da 1 GHz). Una delle tante cose strane che gli appassionati di Windows tendono a esaltare riguarda le DirectX 12. Microsoft ha annunciato che con queste librerie si avranno scenari più ricchi, un maggior numero di oggetti e il pieno utilizzo delle moderne GPU. Naturalmente tutto questo ha delle forti implicazioni per chi ama giocare con il PC. Un ramo, questo, in cui Linux continua a rimanere dietro a Windows. Certo, la situazione sta migliorando da quando Valve ha deciso per il porting di Steam (si hanno circa 1.000 giochi disponibili, tra cui diversi tripla A). Purtroppo, comunque, i numeri continuano a dare ragione a Windows. Solo l'1% degli utenti Steam utilizza Linux (non vengono però riportati i dual-booter). Infatti, sono in molti a mantenere un'installazione di Windows solo per i giochi. Per il Pinguino quello che non manca sono i titoli Indie. Per le piattaforme più evolute, invece, a mettersi di mezzo c'è sempre un porting poco professionale che porta a risultati spesso disastrosi. Il problema, infatti, è la conversione tra DirectX e OpenGL che nella migliore delle ipotesi può essere aggirata utilizzando l'emulatore **Wine**.

> **"Uno degli aspetti più strani dei fan di Windows è la loro propensione a idolatrare le DirectX "**

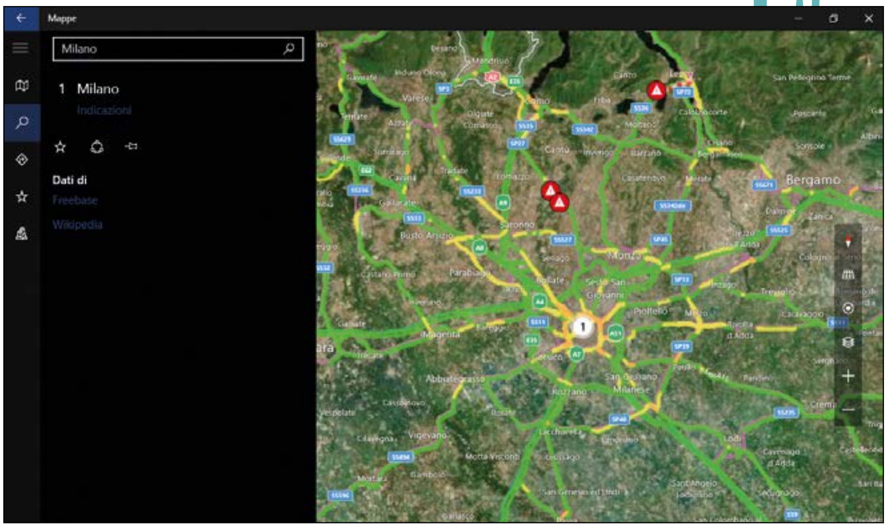
Un altro aspetto che va a discapito del Software Libero consiste nell'uso dei driver proprietari. Chi gioca e vuole ottenere le migliori prestazioni deve per forza rivolgersi a questi componenti. **Mesa**, la controparte FOSS di OpenGL, attualmente supporta solo fino a OpenGL 3.3 che oramai ha compiuto cinque anni. Le versioni più recenti dei driver, invece, supportano la release 4.5 che è stata introdotta circa un anno fa. AMD, grazie alla tecnologia **Mantle**, ha fatto diversi sforzi per rompere la morsa delle DirectX.
Ciò nonostante, anche se il progetto non è stato abbandonato ufficialmente, i fatti ci dicono che il colosso dei processori ha deciso di rivolgere i propri sforzi altrove. Le più recenti tecniche OpenGL, tra cui quella soprannominata *Approaching Zero Overhead Driver* (AZDO), offrono un aumento delle prestazioni simile a quello delle DirectX 12. In fin dei conti, però, il progetto OpenGL sta per compiere 20 anni e prima o poi finirà come il protocollo X e sarà quindi mandato in pensione. I suoi sviluppatori hanno già annunciato **Vulkan**, il suo predecessore. Source 2 di Valve, per esempio, supporta già le nuove librerie, mentre molti altri motori di gioco (CryENGINE, Unity, Unreal, ecc.) sono già supportati da OpenGL. Il numero dei giochi su Linux è quindi destinato a crescere nel medio-lungo termine.



› La serie *Borderlands* è uno dei titoli tripla A disponibili per Linux. Oltre a questo, ci sono *Bioshock Infinite* e *Shadow of Mordor* in arrivo

Desktop & applicazioni

Indipendentemente da quello che Microsoft racconta, Windows 10 non sarà adatto a tutti. Chi proviene da Windows 7 si troverà a fare i conti con le Tiles che, seppure in quantità minore rispetto a Windows 8, sono comunque presenti nel menu Start. Alcune impostazioni di sistema, inoltre, sono difficili da trovare. Il vecchio pannello di controllo è ancora lì ma c'è anche una nuova voce Impostazioni collocata nel menu Start. Questa dualità è presente anche in quest'ultimo elemento che sembra costruito con due sistemi completamente diversi e in parte autonomi: il menu stesso e le Live Tiles a destra. Le app possono poi assere aggiunte in modo un po' goffo da sinistra a destra. In senso contrario, però, non funziona. In generale, riorganizzare le Live Tile è più una questione di gusti casuali che di utilità.



› Windows 10 fa molta pubblicità all'applicazione *Mappe*. Funziona bene e ha un'interfaccia pulita, ma niente che non si sia già visto su Bing con l'omonima funzione

Parlando invece più nello specifico delle applicazioni, al di fuori dell'ambiente professionale, Microsoft Office sta perdendo la sua morsa. La maggior parte delle persone troveranno tutto quello che serve in LibreOffice, mentre i restanti preferiscono lavorare online con Google Docs. Gimp, invece, è più che sufficiente per affrontare la maggior parte delle sfide legate al fotoritocco. Questo soprattutto in virtù del costo di Photoshop. Come utenti Linux, se mai vi troverete di fronte a un file DOC, Google o LibreOffice non potranno aprirlo in modo soddisfacente, quindi avrete bisogno di utilizzare **Office Online** (tramite un Microsoft Account), così da convertirlo in PDF. Molti organismi internazionali hanno comunque scelto **Open Document Format** come standard predefinito. Parecchie aziende desiderose di avere un'ulteriore scusa per passare all'uso della Cloud hanno trasferito praticamente tutti i loro documenti in **Google Docs**. Se tutto va come previsto, quindi, DOC non rimarrà in giro molto ancora. Attraverso Office365 e Creative Cloud, Microsoft e Adobe si stanno sempre più spostando verso la Cloud, modificando quindi le offerte dei loro programmi con piani di abbonamento mirati. Al momento non significa che le loro applicazioni siano disponibili interamente sulla Nuvola, ma ciò non toglie che la direzione futura sia questa. Se ciò dovesse accadere con i dovuti accorgimenti per il rispetto degli standard, allora anche gli utenti Linux potranno beneficiarne, sfruttando così la massima compatibilità. Il software Open Source è migliorato molto in questi anni, tanto da impensierire notevolmente diversi colossi del settore. Inkscape e Krita, per esempio, possono battersi quasi ad armi pari con Illustrator, così come Scribus può farlo con InDesign. Gli utenti di Windows 8.1, seppure abbiano goduto di un nuovo pulsante **Start**, sono rimasti delusi dall'assenza di un vero menu per l'uso delle applicazioni. In Windows 10 sono però stati accontentati. Infatti, il menu **Start** richiama i collegamenti alle app usate più di frequente e a tutta un'altra serie di funzioni. Se poi avete un dispositivo touch, scoprirete come Windows 10 sia piacevolmente pratico da usare. Le due personalità, comandi tattili e mouse e tastiera, sono ancora ben definite ed è proprio per questo che il sistema funziona. Chi è ancora legato ai vecchi dispositivi di input troverà tutto quello di cui ha bisogno e non rimpiangerà l'assenza di uno schermo touch. Al contrario, chi ama usare le dita, sarà accontentato sotto tutti gli aspetti. Gli Hot Corners sono stati aboliti, quindi non sussiste più il problema del puntatore che apriva le finestre indebitamente ogni volta che veniva lasciato su un angolo dello schermo. Nel complesso, quindi, Windows 10 può essere paragonato senza problemi a distro del calibro di Linux Mint con desktop Cinnamon. In altre parole, siamo di fronte a un mix di moderno e funzionale che permette a tutti gli utenti di trovare quello che cercano.

»

Vecchio sistema, nuove funzioni

Per gli utenti che dispongono di computer obsoleti o semplicemente per coloro che non vogliono sprecare soldi in costose periferiche, c'è una grande quantità di desktop leggeri e flessibili. **Xfce**, **LXQt** e **Mate** sono un esempio, ma vale la pena ricordare anche **Pantheon** di ElementaryOS. Ci sono poi **Unity** e **Gnome**, i pionieri dell'ambiente desktop. Riguardo a questi ultimi due, è innegabile come per abituarsi al loro uso ci sia bisogno di un po' di pratica. Ciò nonostante, almeno per quanto riguarda Gnome, si stanno facendo passi avanti per migliorarne l'usabilità. Infatti, adesso basta imparare un paio di scorciatoie da tastiera, adeguarsi al flusso di lavoro e tutto filerà molto più velocemente. In Windows 10 c'è anche un nuovo browser chiamato **Edge** che possiamo definire come IE spogliato del supporto per IE6-ismi. Sembra funzionare bene, anche se la versione provata nella beta ha avuto qualche problema con alcuni siti Web. Seppure le sue potenzialità siano notevoli, sarà comunque difficile strappare gli utenti alla familiarità con cui oramai usano Chrome o Firefox.

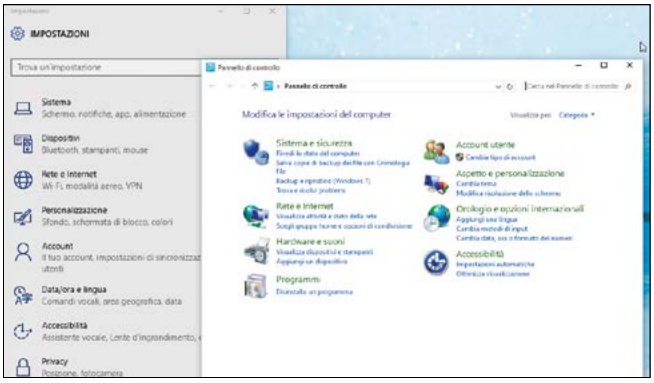
Hardware e driver

Tutti coloro che hanno un vecchio computer o un PC con Windows 7 è importante che tengano a mente i requisiti hardware minimi di 10: CPU 1 GHz, 1 GB di RAM (2 GB per 64-bit), disco rigido da 16 GB e scheda video con supporto per DirectX 9 (con driver WDDM). Considerando le precedenti richieste di Windows Vista, possiamo dire che Windows 10 ha bisogno di molte meno risorse. DirectX 9 è stato sviluppato intorno al 2004, ma l'hardware di quel periodo non riuscirà comunque a soddisfare i requisiti richiesti dalle librerie. Ci sono però parecchie periferiche capaci di farlo, come la serie 600 delle Nvidia GeForce o le HD2xxx AMD del 2006. Naturalmente stiamo parlando dei requisiti minimi.

Se li soddisfate, non aspettatevi comunque un'esperienza d'uso eccezionale. Rallentamenti e colli di bottiglia saranno all'ordine del giorno.

Con un vecchio processore Celeron da 1 GHz, per esempio, ci vorrà parecchio tempo prima di avviare il sistema operativo. Se però avete un PC da ufficio relativamente moderno, vale a dire acquistato all'incirca due anni fa, non dovreste avere grossi problemi. Nonostante le prestazioni reali, è però importante fare onore al merito.

È bello vedere Microsoft che si sforza di contemplare anche chi ha un computer obsoleto e con specifiche hardware di livello basso. Per quanto riguarda il supporto, la situazione è molto migliorata rispetto al passato, ma ci sono comunque diverse periferiche che ancora non riescono a funzionare con Windows 10. Alcuni esempi possono essere diversi chip grafici tra i più vecchi e modificati dai produttori OEM, così come diverse schede TV dotate di telecomando. Anche le schede wireless sembrano essere un punto debole dell'ultimo nato in casa Redmond.



› ***Windows 10*** **dispone di due pannelli di controllo, utili probabilmente ad aiutare chi viene dai vecchi sistemi operativi**

Ci sono molte persone che in questo momento si stanno lamentando sui vari forum per la mancata compatibilità. Tuttavia è importante ricordare come nel 90% dei casi questi malfunzionamenti siano dovuti a un firmware non aggiornato. Un problema che probabilmente si risolverà da solo una volta che il sistema operativo avrà preso piede nella distribuzione di massa. Tutti questi inconvenienti, così come quelli relativi ai driver, non si riscontrano in Linux, dove la maggior parte delle periferiche è supportata a livello di kernel. Una volta caricati automaticamente i rispettivi moduli, basterà qualche aggiustamento nella configurazione, ma sarà molto raro dovervi poi rimettere mano o addirittura pensare di sostituire l'hardware per mancanza di supporto. C'è poi tutta la parte relativa ai driver proprietari, ma anche in questo caso non ci sono problemi, se non per coloro che ovviamente sono legati a una filosofia puramente Open e quindi mal digeriscono qualsiasi software che non sia aperto. Comunque, in questo caso, si fa quasi sempre riferimento alle schede grafiche AMD e Nvidia che peraltro forniscono senza grossi ritardi i driver Linux adatti alle loro periferiche. Dall'altra parte, in Windows è molto facile che si presentino problemi legati ai driver, soprattutto a causa delle specifiche richieste da Microsoft per la compatibilità e quindi per la loro firma.

Una scheda madre, per esempio, richiede driver per il chipset, l'interfaccia di rete, il controller RAID, la periferica audio e molto altro. Per l'hardware abbastanza recente, potrete trovare tutto sui siti dei produttori. Tuttavia avrete bisogno di sapere con estrema certezza di quali versioni avrete bisogno. In più, dovrete sobbarcarvi un buon lavoro di download, dato che nella maggior parte dei casi parliamo di pacchetti pesanti diverse centinaia di MB. Se non avete una buona connessione, quindi, rischiate di lasciare il PC acceso la notte per scaricare tutto. I driver Linux, al contrario, sono sottoposti al vaglio dei vari manutentori di sottosistemi e sono garantiti per essere efficienti e ben codificati.

> **"I neofiti del mondo Linux, commettono spesso l'errore di andare a caccia di driver"**

Driver grafici

Le schede grafiche più recenti tendono a funzionare meglio su Windows anziché su Linux. Tuttavia, sebbene ci voglia un po' di pazienza, riescono a restituire ottimi risultati anche sul nostro sistema. Di solito, i driver per Windows vengono messi a disposizione immediatamente prima del lancio di un nuovo prodotto. Detto questo e volendo portare un esempio concreto, possiamo parlare di Nvidia. Il nuovo hardware di questo produttore richiede un firmware firmato in ogni sua parte e sembra che il colosso delle schede grafiche abbia qualche difficoltà a fornire il necessario al progetto Noveau Open Source. AMD, invece, si dimostra molto più amichevole con il progetto Radeon Open Source. Non solo fornisce le specifiche, ma paga del personale per occuparsene. L'ultima innovazione di AMD, per esempio, è stata quella di introdurre un modulo comune del kernel sia per i driver Catalyst proprietari sia Open Source.



Dietro al desktop

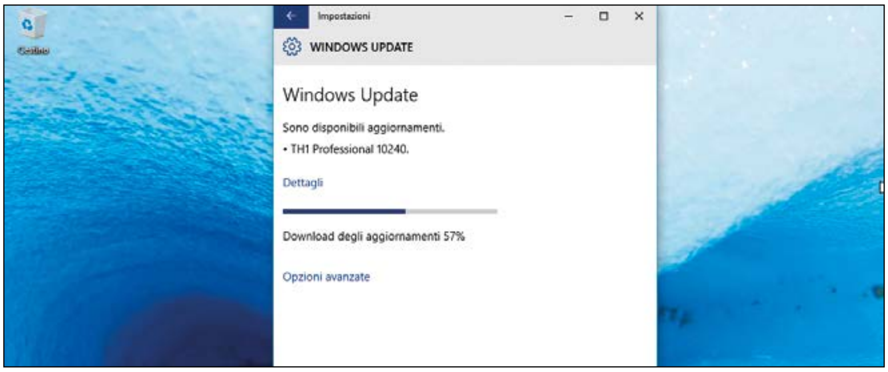
La prossima edizione di Windows Server non sarà rilasciata fino al prossimo anno. Tuttavia ci sono alcune anteprime disponibili. La grande novità consiste nell'*Active Directory Federation Services* (ADFS) che permette agli utenti di directory e database diversi di essere autenticati da domini Active Directory. ADFS stesso è stato parte del sistema operativo Windows Server 2003 R2 e consente agli utenti che provengono da altri sistemi di usare le proprie credenziali senza alcun problema. Esistono già soluzioni commerciali che permettono di autenticare i clienti Linux con un controller di dominio Active Directory. In più, è possibile farlo utilizzando software FOSS, anche se è bene ricordare che non si tratta di una procedura semplice da mettere in pratica. Active Directory, infatti, utilizza gli standard aperti LDAP e Kerberos. Questi, però, devono essere uniti da Samba e PAM.

Nella nuova edizione, tale processo dovrebbe comunque rivelarsi più immediato e flessibile. L'autenticazione centralizzata in un ambiente Linux puro può essere eseguita tramite i protocolli di cui sopra o in alternativa con SASL o NIS. Tutti questi approcci hanno però vantaggi e svantaggi e coloro che provengono da un background Microsoft possono avere più difficoltà nel ricreare le funzioni avanzate di Active Directory. È poi importante notare come Active Directory stesso fornisca un'autenticazione più semplice, gestendo poi tutti gli elementi in modo flessibile (certificati, domini, criteri di gruppo, ecc.). Molti di questi vengono rilevati solo su sistemi Windows, mentre il resto può essere affrontato con l'utilizzo di altri strumenti Linux. La versione 10 di **Internet Information Services** (ISS) è stata inclusa in Windows 10, portandosi dietro anche il supporto per HTTP/2. Naturalmente, i nostri primi tre Web server Linux (Apache, Nginx e Lighttpd) hanno iniziato a supportare questo protocollo dopo RFC7540 (pubblicata nel mese di maggio). Prima della versione 7.0, ISS poteva essere considerato un server giocattolo, eccessivamente pesante e incapace di permettere più di 10 connessioni simultanee.

Il progetto è comunque molto cresciuto e migliorato, offrendo adesso un sistema di estensione modulare che consente un'ottima scalabilità su sistemi multiprocessore.

Per perfezionare le proprie prestazioni, ISS utilizza un driver a livello di kernel per l'elaborazione delle richieste HTTP. Una vulnerabilità del server scoperta nel mese di aprile ha però permesso a degli aggressori di eseguire del codice remoto su sistemi privi di patch. Il tutto sfruttando il driver di cui sopra. I server Web Linux non sono indenni da bug, ma nessuno di questi ha mai preso in considerazione il kernel.

In altre parole, Linux rimane il campione indiscusso in ambito server ed è per questo che gestisce la maggior parte dell'infrastruttura Internet odierna. Abbiamo server Web di classe mondiale, banche dati, distribuzioni di livello professionale (Red Hat Enterprise Linux o CentOS) e il vantaggio dell'Open Source. Le macchine virtuali Linux, inoltre, tendono a essere molto più economiche rispetto alle loro omologhe Windows e sono molto più efficienti grazie alla loro natura modulare. LXP



› **Per aggiornare Windows, basta accedere a *Windows Update***

> **"Prima della versione 7.0, ISS poteva essere considerato un server Web per principianti"**

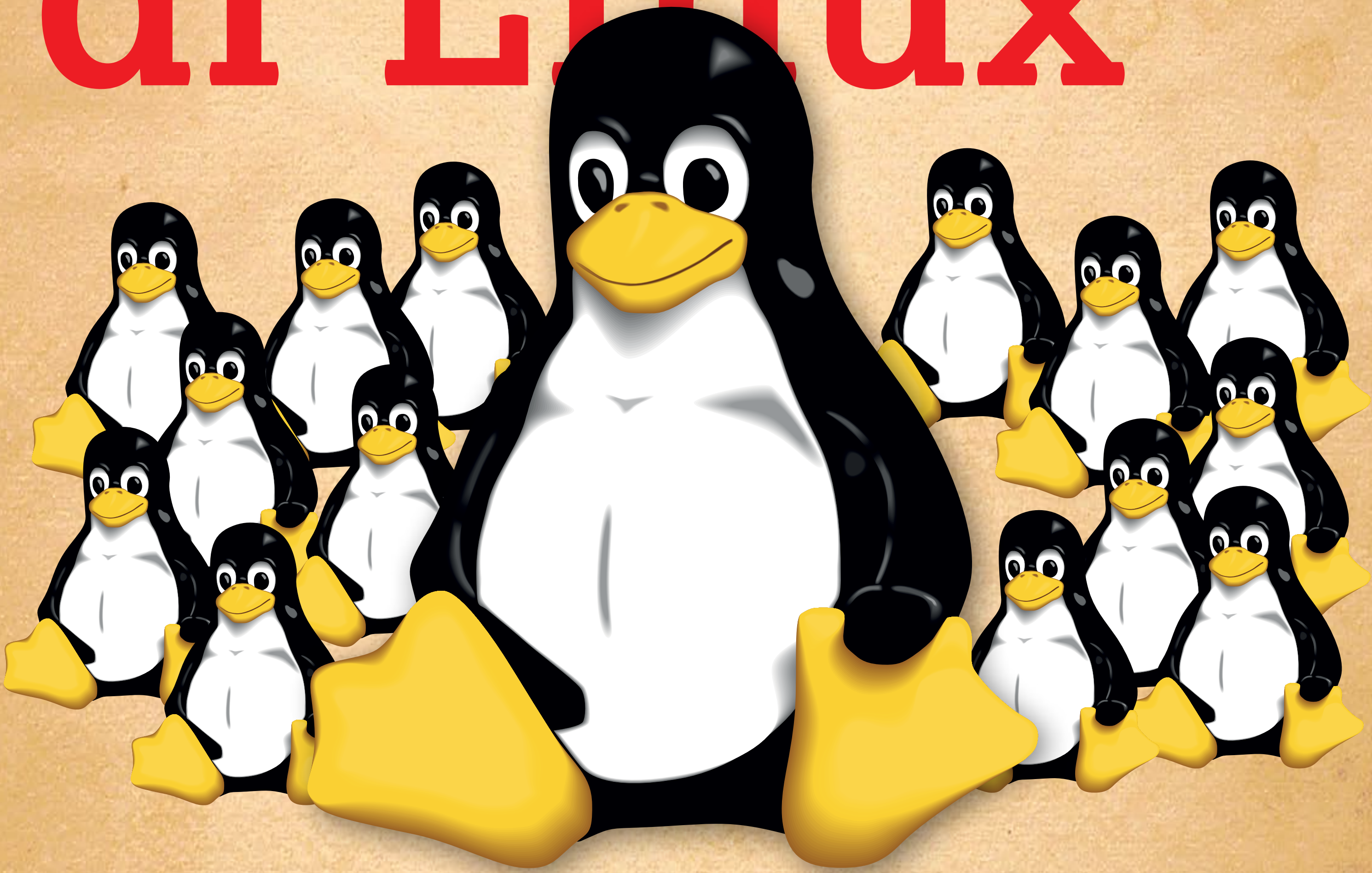
Windows IoT Pi Edition

Windows Server Core è stato introdotto nel 2008 e ha fornito un ambiente minimale con OS e shell, più una serie di altre caratteristiche non necessariamente richieste. Continuando su questo tema, Microsoft ha deciso di creare **Windows IoT** destinato all'Internet delle cose. Allo stato attuale, è disponibile per cinque dispositivi, tra cui il Raspberry Pi 2. Naturalmente, questo non significa che potrete usare Edge e le Live Ties in un desktop Pi. Per la verità, non si tratterà neppure di un Pi, dato che tutto il codice è stato scritto in Visual Studio su una macchina Windows 10 e caricato in Pi. Il progetto è stato sviluppato per consentire di usare le Universal App che però devono essere scritte in C#, C++, JavaScript, XAML o HTML. È poi possibile connettersi a un Pi che esegue Windows IoT tramite PowerShell o SSH. In questo caso, tuttavia, non riusciamo proprio a essere imparziali. Raspbian è sicuramente da preferirsi su tutti i fronti. Ciò nonostante, va fatto un plauso all'apertura di Microsoft nel considerare anche i micro-computer che permettono ai programmatori di affrontare il proprio lavoro su dispositivi pensati per loro. Chi si occupa di questo genere di attività in modo professionale, continuerà quasi sicuramente a usare le tante distro disponibili, soprattutto anche in virtù della



documentazione presente in Rete. Se però volete cimentarvi in un'esperienza diversa, niente vieta di dare uno sguardo anche a questa nuova versione di Windows.

15 anni di Linux

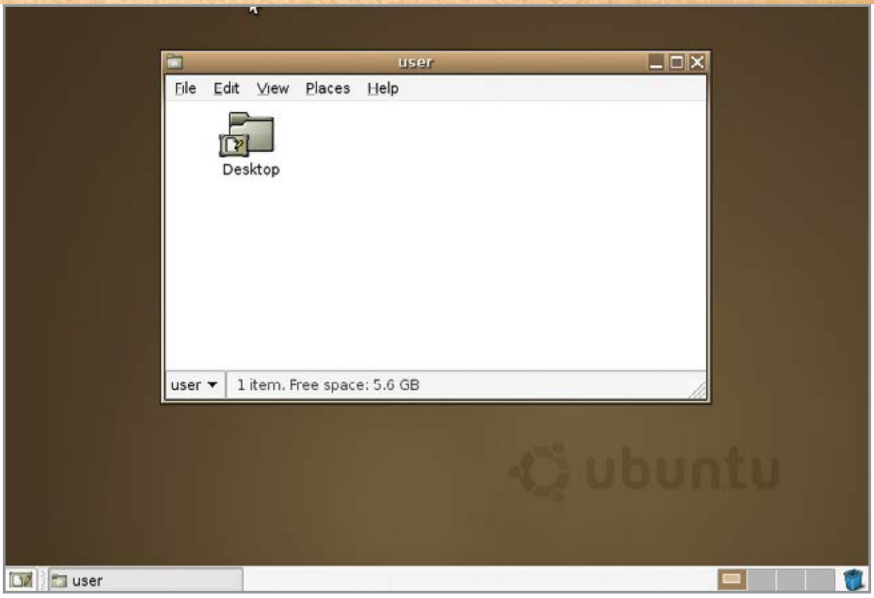


In questo millennio Linux ha cambiato la storia dei sistemi operativi: dal 2000 a oggi ripercorriamo 15 anni di successi...

Sembra impossibile, ma Linux ha 24 anni. È nel pieno della sua gioventù e proiettato verso un futuro radioso ma non è più un bambino... e neanche un adolescente. Se i suoi esordi, con un giovane Linus Torvalds all'università di Helsinky nei primi anni '90, come la nascita del suo nome, che originariamente doveva essere Freax, termine creato dall'unione delle parole inglesi Free (libero) e Freak (fenomeno da baraccone) con l'aggiunta di una x finale (per ricordare Unix), e l'adozione del pinguino Tux (Torvalds Unix) come mascotte ufficiale nel 1996, vengono spesso ricordati, sono in pochi a essersi soffermati sull'avventura di Linux nel nuovo millennio. Partiamo quindi insieme in questo viaggio nel passato per vedere come Linux è diventato l'incredibile fenomeno di oggi.

> **"Facciamo un viaggio nel terzo millennio per vedere come Linux è diventato il fenomeno di oggi"**

All'alba del nuovo millennio, Linux aveva già nove anni. Alcune distribuzioni (distro) si erano già affermate e una delle prime era stata Softlanding Linux System (SLS), pubblicata nel maggio del 1992. A differenza di altre distro dello stesso periodo, SLS offriva più che il kernel e qualche strumento GNU per preparare i filesystem: aveva anche un networking stack e il display server X. Era considerata tanto ambiziosa quanto ricca di bug e gli sforzi per sistemarla sono culminati nella pubblicazione di Slackware nel 1993. Nello stesso anno, sempre come risposta alle difficoltà di SLS, è nato Debian. L'anno dopo è comparso Red Hat Commercial Linux, che ha portato all'estinzione molte distro popolari alla fine degli anni Novanta, come Mandrake, Yellow Dog e Definite Linux. KDE è uscito nel 1998, seguito da Gnome nel 1999. Uno dei motivi che hanno portato alla creazione di Gnome è stato il fatto che KDE si basava molto sugli strumenti *Qt*, che all'epoca non erano freeware. Nel 2000, le distribuzioni più popolari erano Debian 2.1, Red Hat 6.1, Linux-Mandrake 7.0 (allora si chiamava così), Slackware 7.0 e SUSE Linux 6.3.



› Il tema Human ha rappresentato un tentativo di rendere Ubuntu più piacevole per gli utenti. Come tutti sanno, infatti, il marroncino è un colore meraviglioso, soprattutto se sei un facocero

In che senso user experience?

Se vi siete convertiti recentemente a Linux e vi siete trovati a combattere con file di configurazione ribelli, driver dispettosi o altre problematiche affini, pensate agli utenti degli esordi di Linux. Grazie ai loro bug report e invettive assortite siamo faticosamente arrivati al desktop di Linux come lo conosciamo oggi. Non dimentichiamoci che in tempi relativamente recenti era ancora possibile distruggere il monitor dando informazioni temporali sbagliate a X. Avete mai avuto problemi con **Grub**? Provate a immaginare di dover padroneggiare una delle prime versioni di **Lilo**... Nei primi anni di Linux, neanche far funzionare un mouse era un'operazione triviale, perché l'utente doveva fare una serie di calibrazioni manuali. Red Hat poi ha pubblicato uno strumento chiamato **Xconfigurator** che offriva un'interfaccia testuale a menu per impostare il server X. È stata accolta come la manna dal cielo, anche se la sua unica funzione era generare un file **XF86Config** che altrimenti gli utenti avrebbero dovuto scriversi da soli. Così, mentre gli utenti di Windows ME si lamentavano che era un po' lento e disabilitava la modalità DOS, l'utente Linux medio saltava di gioia se riusciva a completare il processo di installazione.

Una volta superata questa fase, inoltre, sarebbe stato troppo ottimistico aspettarsi che il boot del sistema operativo funzionasse. Le funzioni di riconoscimento dell'hardware sostanzialmente non esistevano e la maggior parte dei pochi driver scritti per Linux non era di qualità professionale. Eppure, nonostante tutte le difficoltà, i pionieri hanno perseverato. Molti preferivano lavorare in DOS che ha iniziato a essere messo da parte con l'avvicinarsi della fine del millennio. Nel gennaio del 2001 è stato pubblicato il Kernel 2.4 e con esso sono arrivati il supporto per i dispositivi USB e gli entusiasmanti nuovi processori Pentium IV, tra le altre innovazioni del periodo. È stato particolarmente importante per chi aveva un desktop grazie al suo trattamento unificato dei dispositivi PCI, ISA, PC Card e PnP, oltre che per il supporto ACPI. La bolla speculativa delle "dot-com" sul mercato finanziario stava per scoppiare ma tutto l'entusiasmo e la speculazione che lo circondavano ha fatto sì che molti appassionati di computer potessero avere una connessione a banda larga a casa e alcuni si godevano persino il lusso di possedere più di un computer. In questo modo sono sparite delle importanti barriere che ostacolavano l'avvicinamento al mondo Linux: ora la gente poteva scaricarlo molto più facilmente, non era difficile accedere a della documentazione aggiornata e, quando Linux decideva di far sparire la connessione a Internet del computer o bloccare il boot del sistema, si poteva sempre usare il secondo computer per cercare una soluzione. Nel complesso, però, l'ambiente Linux era ancora decisamente inospitale per l'utente. Se alcuni installer avevano introdotto funzionalità grafiche, nella maggior parte dei casi creavano più problemi che benefici. Gli utenti dovevano conoscere le dinamiche del partizionamento del disco da cima a fondo ed essere in grado di capire quali pacchetti dovevano installare sulla sola base delle loro, spesso scarne, descrizioni. **Windows XP** è stato pubblicato nell'ottobre del 2001 e, anche se è stato accolto come un enorme miglioramento rispetto al suo predecessore, molti utenti hanno avuto l'impressione che i loro computer non avessero la potenza per farlo girare. Dopo tutto, richiedeva 64 MB RAM e ben 1,5 GB di spazio su disco. Va ricordato che i BIOS hanno raggiunto solo recentemente la capacità di gestire drive di grandi dimensioni (i limiti erano diversi in base al tipo di BIOS, ma 2,1, 4,2 e 8,4 GB erano barriere comuni). Di conseguenza molti utenti non lo potevano installare sul loro hardware e anche chi riusciva a raggiungere le specifiche minime vedeva le prestazioni del computer degradare rapidamente una volta che tutte »

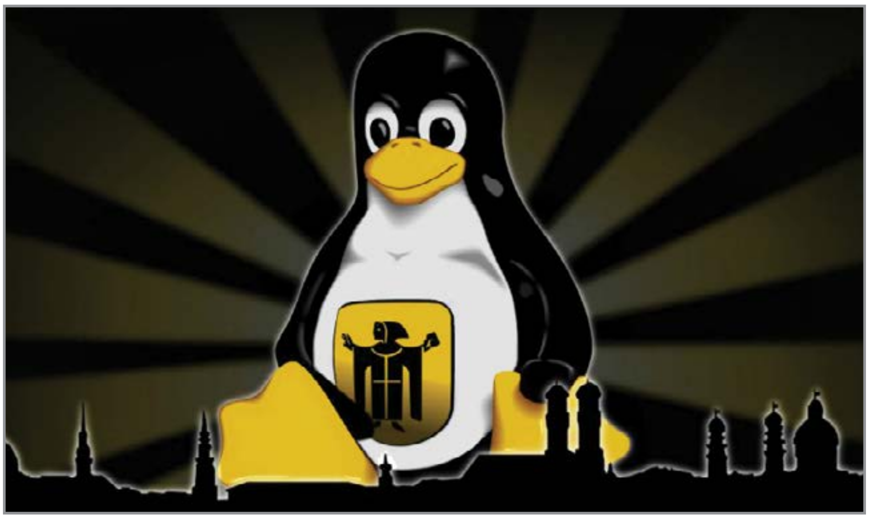
> "Neanche far funzionare un mouse era triviale: l'utente doveva fare una serie di calibrazioni manuali"

» le librerie di runtime e le suite di Office erano installate. Queste difficoltà hanno causato un altro piccolo esodo verso Linux e gli utenti che non hanno un hardware molto aggiornato continuano ancora oggi a rappresentare una parte significativa della base di utenti di Linux (a volte i lettori si lamentano che non includiamo distro a 32-bit nel DVD di Linux Pro). Prima del 2006 tutti i Mac avevano processori PowerPC e molti di essi (come anche alcuni dei primi Mac Intel), a lungo deprivati di aggiornamenti software da parte di Apple, oggi adottano Linux. L'ambiente desktop Gnome 2 è stato pubblicato nel 2002 ed è diventato così significativo che alcuni utenti ancora oggi (per nostalgia, affetto o insoddisfazione riguardo alle alternative moderne) cercano di riprodurlo. Si proponeva di essere semplice, modificabile e intuitivo e non si può proprio dire che non abbia raggiunto al meglio questi obiettivi.

Belli e possibili

Uno dei suoi punti di forza più significativi era la notevole aderenza al set di norme per l'interfaccia utente di Gnome, che afferma alcuni principi chiave per gli sviluppatori di applicazioni. Il desktop di conseguenza non era solo coerente a livello interno ma anche con tutte le applicazioni **GTK** che si potevano scrivere per sfruttarlo. È stato anche pubblicato **KDE 3**, che somigliava vagamente a Windows, sotto il profilo estetico e per il fatto che richiedeva risorse di sistema più impegnative di quelle di Gnome. Gli utenti e le distro si sono schierati da una parte o dall'altra. **SUSE Linux** (predecessore di openSUSE) ha sempre avuto come obiettivo quello di essere slegato dalla scelta del desktop, ma la maggior parte dei suoi utenti preferiva KDE. A causa di questa propensione ha cambiato orientamento (anche se ha aspettato fino al 2009 per farlo) e oggi è la principale distro per KDE. Alla fine del 2002, 'DVD' Jon Johansen è stato oggetto di una causa per la pubblicazione nel 1999 del software **DeCSS** per aggirare il sistema di protezione *Content Scrambling System* (CSS) usato dai DVD commerciali. Questo software permetteva agli utenti Linux di usare i DVD, obiettivo precedentemente irraggiungibile perché il sofware dei DVD richiedeva una chiave di licenza della DVD Copy Control Agency, una delle aziende a intentare la causa. Si è poi appurato che il CSS si poteva violare in modo molto più triviale e alla fine Johansen non è stato condannato. In questo periodo gli iPod e la pirateria avevano portato a una larghissima diffusione dei file MP3. A questo tipo di file era (e ancora è) collegata una serie di diatribe sui brevetti, con diverse entità che rivendicavano la proprietà di varie parti degli algoritmi su cui si basa. Di conseguenza, molte distro sono uscite senza codec multimediali su cui potevano gravare problemi di brevetto. La legge però non è chiara a questo proposito e i detentori dei diritti si sono mostrati poco inclini a fare causa alle implementazioni di questi codec da parte di Software Libero. La maggior parte delle distro adotta un approccio prudente e lascia agli utenti la scelta di installarli o meno, anche se Ubuntu propone i codec Fluendo (su licenza ma proprietari) all'installazione. Fortunatamente, molti dei brevetti degli MP3 sono scaduti, e molti altri lo faranno entro il 2017, ma oggi non ha una grande rilevanza; ci sono numerosi formati e codec open (OGG, FLAC, VPx e x264). Tecnicamente è ancora una violazione DMCA usare **libdvdcss** (un metodo moderno e molto più efficiente di aggirare CSS, usato dalla maggior parte dei lettori multimediali su Linux) per guardare i DVD, ma la restrizione si applica solo ad alcuni Paesi e fino a oggi nessuno vi ha mai fatto appello. La città di Monaco di Baviera ha annunciato nel 2003 che avrebbe fatto migrare tutte le sue infrastrutture da Windows NT a Linux. Oltre che dall'ovvio risparmio, le autorità locali hanno affermato che la scelta era motivata dal voler evitare la dipendenza da un fornitore. Steve Ballmer ha fatto visita personalmente al sindaco della città ma persino il suo fascino e la sua eloquenza (unite, presumibilmente, alle sue offerte di importanti sconti) non sono riuscite a far cambiare idea ai rivoluzionari. Il progetto è stato completato dieci anni dopo con la transizione di circa 15.000 computer alla distro personalizzata **LiMux**. Nel 2014 si è diffusa la voce che la città sarebbe tornata a Windows, ma era una diceria infondata. Si stima che il cambiamento abbia permesso a Monaco di risparmiare circa 11 milioni di euro.

> **"Gnome 2 è stato così significativo che alcuni utenti ancora oggi cercano di riprodurlo"**



› **Il progetto *LiMux* ha fatto a Tux un "tatuaggio" con l'emblema di Monaco, il Münchner Kindl. Ci assicurano che il pinguino non ha sofferto e il progetto ha permesso alla città di risparmiare circa 11 milioni di euro**

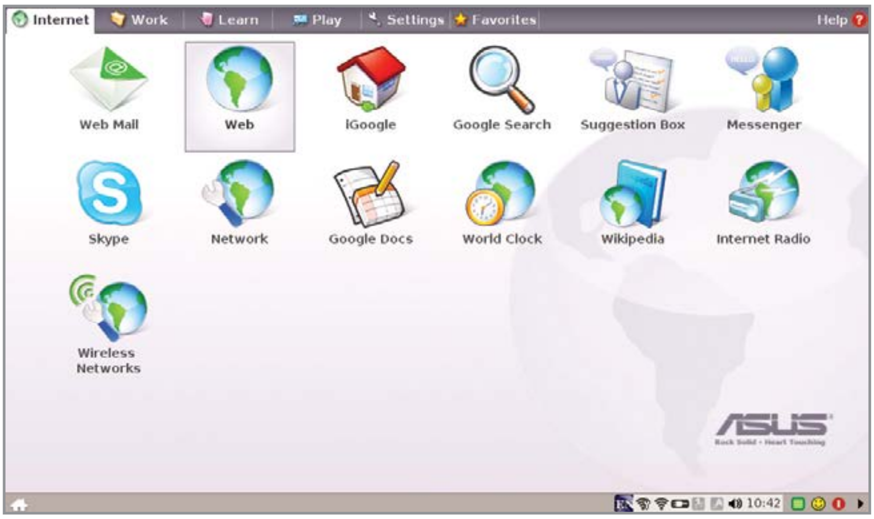
Il kernel del cuore

Nel 2003, dopo due anni di sviluppo, è stato pubblicato il Kernel 2.6. Era una completa rivoluzione rispetto al 2.4, con miglioramenti dello scheduler, maggior supporto per i sistemi multiprocessore (inclusi hyperthreading, NPTL e NUMA), I/O più veloce e molto più supporto per l'hardware. È stata introdotta anche la *Physical Address Extension* (PAE) per permettere alle macchine di gestire fino a 64 GB di RAM anche con un'architettura a 32 bit. Un'altra caratteristica nata con questo kernel è il venerabile sottosistema *Advanced Linux Sound Architecture* (ALSA) che permetteva di usare (quasi) istantaneamente le schede

I driver grafici e le loro disavventure

Nel 2003 sia Ati (ora parte di AMD) sia Nvidia avevano pubblicato driver proprietari per sfruttare al meglio le capacità 3D del loro nuovo hardware (in quel periodo le schede di punta erano la X1800 e la serie 6800 di Nvidia). Erano disponibili dei driver Open Source ma le prestazioni non erano brillanti. ATI era molto più disponibile dei suoi concorrenti a rendere note le specifiche e di conseguenza i loro driver Open Source si sono sviluppati molto più rapidamente. Nvidia, tramite il suo nv driver, ha pubblicato solo del codice sorgente offuscato che ha lasciato gli sviluppatori perplessi e frustrati. I driver binari erano problematici, anche con l'aiuto di utili strumenti di gestione come **Jockey** di Ubuntu. Le Repository non riuscivano a tenere il passo con le nuove uscite, il che ha portato gli utenti a scaricare i pacchetti direttamente da AMD o Nvidia. Questi però davano dei problemi (non ci piacciono troppo neanche ora) e facevano disastri con i driver esistenti. Dato che non erano compresi nella gestione del package manager, tutte le volte che c'era un update del kernel bisognava ricompilare il modulo dei driver, altrimenti al riavvio successivo l'utente si sarebbe trovato senza alcuna grafica, fenomeno che comprensibilmente causava un certo scontento. Questa specifica situazione è stata migliorata grazie a DKMS ma i problemi grafici continuano a essere fonte di frustrazione per molti utenti. Le difficoltà rimangono sotto molti aspetti le stesse: i driver Open Source sono lenti e quelli binari causano disastri assortiti. In risposta alle prestazioni scarse e all'assenza di supporto 3D per i driver nv nel 2006 è stato annunciato il progetto **nouveau**. Si trattava di uno sforzo di dimensioni ciclopiche di reverse-engineering con il modello di Brewer e Nash (chiamato anche Chinese Wall) e studiato per evitare il rischio di violazione di copyright. Nouveau si basa in parte su dati forniti dalla comunità: i partecipanti scaricano il programma **REnouveau** che esplora registri e grafica creandone un'istantanea per l'analisi degli sviluppatori. Solo nel 2012 ha raggiunto una release stabile ma è comparso in alcune distro già tre anni prima, perché malgrado i bug erano comunque superiori a nv.

audio più popolari, oltre a offrire supporto per più dispositivi, mixer hardware, full-duplex e MIDI. La novità più significativa, però, è stata la sostituzione del vecchio sottosistema di gestione dei dispositivi, **devfs**, con il nuovo **udev**. È comparso solo dalla versione 2.6.13 (novembre 2003), e a quel punto la directory **/dev** ha smesso di essere una lista infinita di nodi statici ed è diventata una rappresentazione dinamica dei dispositivi effettivamente connessi al sistema. Il gestore di dispositivi udev supportava anche il caricamento di firmware ed eventi nello spazio utente e ha contribuito a creare un ambiente molto più pratico per gli utenti di desktop, anche se bisognava ancora fare affidamento sui misteri di HAL e ivman per effettuare l'automount di uno stick USB con i permessi giusti. Linux, che era già stato portato su processori non x86 64 bit, supportava l'istruzione di Itanium IA64 quando è stato pubblicato nel 2001. Questa architettura era però destinata a fallire e Intel alla fine è passata alla più conservativa AMD64 (o x86-64) che, ci fa piacere ricordarlo, è in circolazione dal 2003. Grazie al software Open Source, gli utenti Linux hanno avuto subito desktop a 64 bit, mentre chi usava Windows ha dovuto aspettare fino al 2005 per avere la versione x64 di XP. Varie applicazioni proprietarie (tra cui **Steam** e i suoi giochi) girano in modalità 32-bit, il che porta i distributori a mantenere delle release a 32 bit ma verrà il giorno in cui non saranno più sostenibili sotto il profilo economico e alla lunga faranno la fine del 386, non più supportato in Linux dal 2013.



› ***EeePC Linux* di Asus era basato su *Xandros* e *IceWM* ma ai neofiti non è piaciuto e i professionisti lo hanno sostituito**

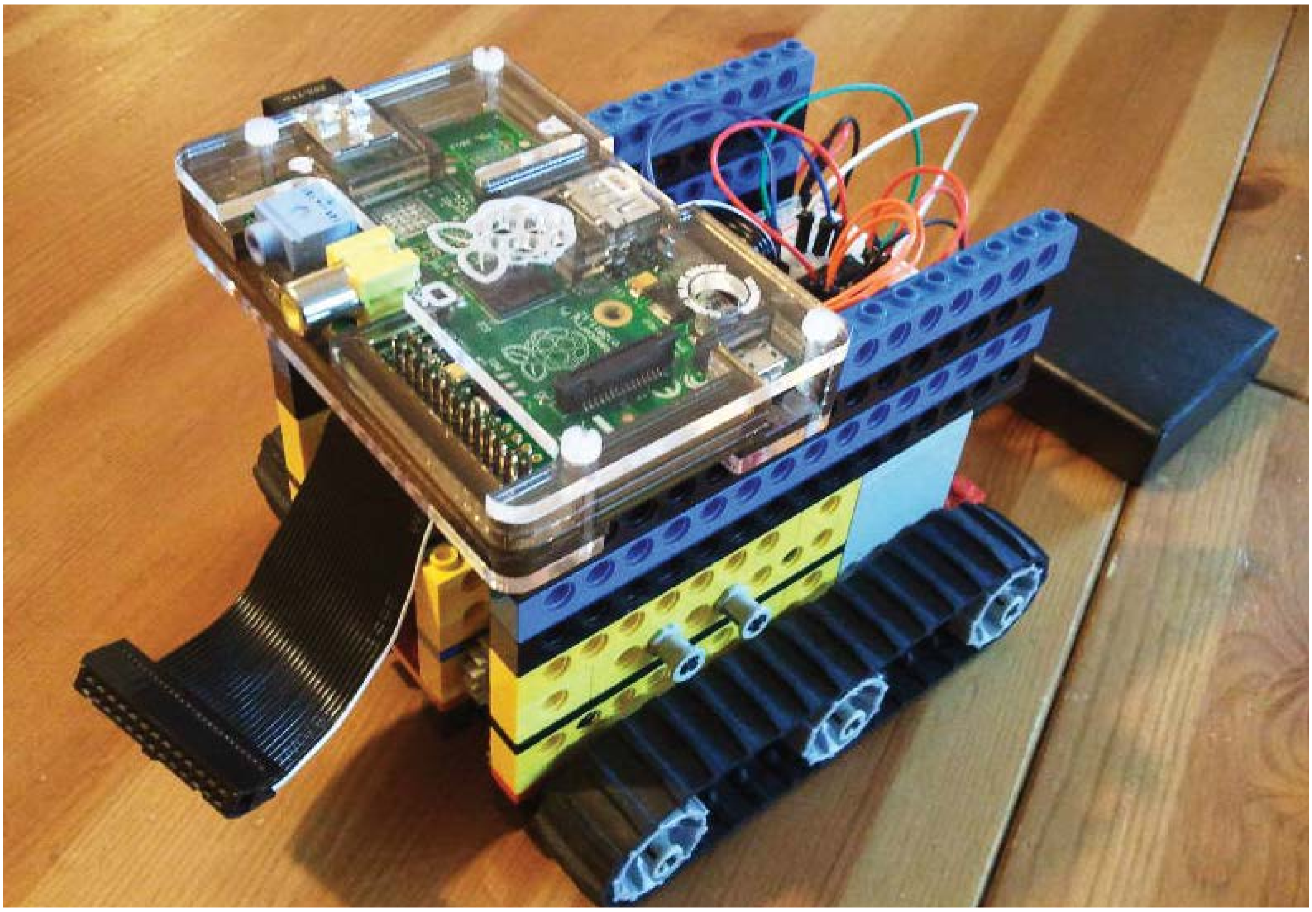
L'esordio dell'archetipo

L'uscita di Ubuntu 4.10 (**Warty Warthog**) nel 2004 è stata senza alcun dubbio una spinta fondamentale per Linux sui desktop. Usando la fortuna ricavata dalla creazione e dalla vendita di Thawte, Mark Shuttleworth ha fondato Canonical Inc. Lo scopo era di vendere prodotti e supporto per i server e allo stesso tempo creare un desktop Linux "per esseri umani". Usando come base Debian, che a questo punto aveva dimostrato il suo valore, Canonical ha introdotto dei miglioramenti per i driver, un tema Gnome 2 molto marroncino e un ambizioso programma di release semestrali. Abbiamo anche visto il lancio di **http://ubuntuforums.org**, dove ben intenzionati ma mal informati membri della community internazionale postavano "soluzioni" più o meno efficaci a vari problemi di Ubuntu. Nel 2004 è uscito il server audio **Polypaudio**, realizzato da uno sviluppatore allora sconosciuto di nome Lennart Poettering e alcuni collaboratori. In quel periodo gli ambienti desktop sfruttavano i server audio per superare i limiti del sistema dmix di ALSA: Gnome usava **Enlightened Sound Daemon** (ESD) e KDE si appoggiava all'analogico **Realtime synthesizer** (aRts). Polypaudio era studiato per sostituire ESD, fornendo funzioni molto più avanzate, come controllo del volume separato per le varie applicazioni e trasparenza in rete. Nel 2006 il progetto ha cambiato nome in **PulseAudio**, citando il fatto che "polyp" non piaceva perché nessuno vuole avere i polipi (anche se il sistema prendeva nome dagli animali marini e non dalla malattia). Con il nuovo nome e la crescente richiesta del mercato per un sistema audio comparabile a quello di OSX o dell'appena pubblicato (e molto malignato) **Windows Vista**, PulseAudio »

» ha visto un notevole sviluppo e ha iniziato a essere preso in considerazione per l'inclusione in varie distro. Come da tradizione, Fedora è stata la prima ad adottarlo, incorporandolo come sistema audio standard nella versione 8, uscita alla fine del 2007. Ubuntu ha seguito il suo esempio nella versione 8.04, anche se la sua implementazione è stata accolta con molte critiche e ha scatenato una serie di considerazioni al vetriolo contro Pulse. Poettering a un certo punto è arrivato a descrivere la sua stessa creazione come: "Il software che attualmente rovina il vostro audio". Ci è voluto un po' di tempo ma alla fine Ubuntu (e con lui altre distro) ha risolto i problemi di implementazione di Pulse e ora funziona senza quasi intervento da parte dell'utente. Prima dei tablet e degli smartphone a un prezzo accessibile anche ai comuni mortali, i netbook erano la punta di diamante dei computer portatili. Il primo è stato l'Asus EeePC 701. A causa delle sue specifiche hardware piuttosto ridotte (aveva un processore a 700 MHz, un display 800x480 e 512 MB di RAM) non era possibile adottare Windows come sistema operativo. È invece uscito con una versione customizzata di Xandros Linux che era funzionale ma mancava un po' di finezza e dettagli. Nell'insieme la maggior parte degli utenti ne è rimasta insoddisfatta ma i netbook si sono comunque dimostrati delle

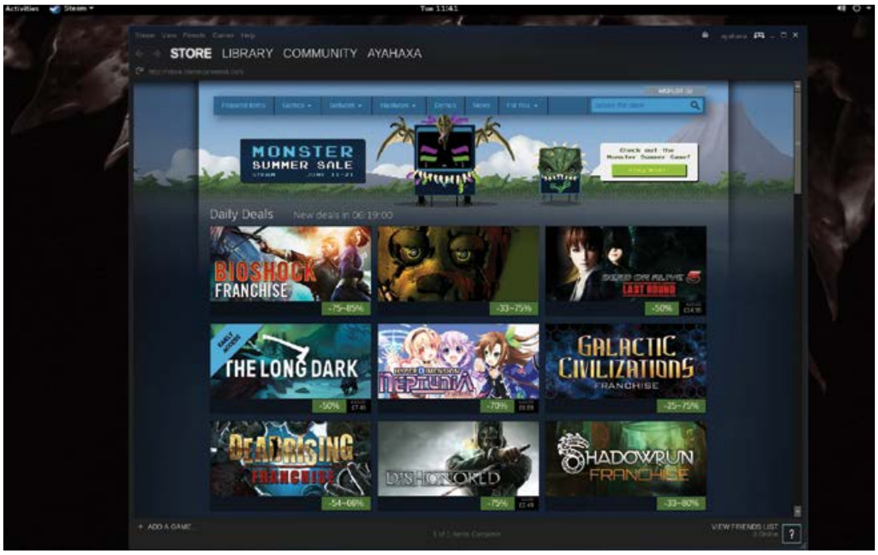
La rivoluzione di Raspberry Pi

Raspberry Pi è stato pubblicato nel 2012. Inspirato in parte dal successo di BBC Micro (microcomputer lanciato da BBC per favorire l'alfabetizzazione informatica) all'inizio degli anni Ottanta, Raspberry Pi mirava a portare l'applicazione pratica dell'informatica nelle classi scolastiche e a coinvolgere l'industria elettronica del Regno Unito. Il computer a basso costo delle dimensioni di una carta di credito ha venduto più di cinque milioni di unità. Molte sono alla base del lavoro di giovani programmatori, ma altrettante sono diventate parte di vari progetti personali di appassionati: la generazione che si è fatta i denti su BBC Micro, Spectrum e C64 rivive le gioie della programmazione della creatività. I connettori GPIO di Raspberry Pi hanno consentito lo sviluppo di infiniti add-on e Pi gestisce di tutto, dai robot ai sistemi di irrigazione a distanza. La lingua franca dei progetti per Raspberry Pi è il Python che, come il Basic, è facile da imparare. A differenza del Basic, però, è coerente, estensibile e gli utenti che vorranno passare a linguaggi più avanzati non avranno bisogno di dimenticarselo. Il supporto per la grafica 3D di Pi è ottimo ma sul fronte CPU è più limitato. I Rasperry Pi originali fanno fatica a operare come computer desktop, anche con la modesta distro Raspbian (anche se il recente lavoro sul Web browser **Epiphany** ha migliorato questo aspetto). Nel 2015 Raspberry Pi è stato rinnovato, con un processore quad-core e più RAM, così adesso è davvero un computer multi funzione a meno di trenta euro.



› ***Raspberry Pi*** **ha ispirato una nuova generazione di creativi. Quando i robot insorgeranno e conquisteranno il mondo potremo dare la colpa alla Raspberry Pi Foundation e alle sue mille ramificazioni**

ottime piattaforme per gli utenti Linux con maggiore esperienza. Man mano che uscivano nuovi modelli di netbook (molti basati sui processori Intel Atom, meglio adatti allo scopo) i produttori hanno cominciato ad adottare Windows XP (circa sette anni dopo la sua pubblicazione....) e infine la scarnificata versione **Windows 7 Starter Edition**. Asus in seguito ha fatto marcia indietro sul suo entusiasmo per Linux: dopo aver stretto una partnership con Microsoft ha persino lanciato una campagna "È meglio con Windows" mirata a scoraggiare gli utenti dal comprare laptop che usavano Linux come sistema operativo. Questa campagna anti-Linux usava frasi come: "Sostanziali problemi di compatibilità" e "Ambiente di lavoro non familiare" per spaventare e allontanare i potenziali utenti.



› Grazie a ***Steam on Linux***, **gli appassionati di videogiochi hanno finalmente centinaia di titoli con cui cimentarsi e i redattori di Linux Pro possono divertirsi con la scusa di lavorare**

Il prezzo del progresso

C'è chi ricorda il 2010 come l'anno in cui Ubuntu ha iniziato a perdere la testa. Fino ad allora, la distro era passata di punto di forza in punto di forza, aumentando la sua base di utenti e guadagnando stabilità. Era l'incarnazione del sogno di portare Linux sui desktop. Nella release 10.10, però, qualcosa ha iniziato ad andare storto. L'**Ubuntu Software Center** includeva applicazioni pagate (la prima è stata il lettore di DVD di Fluendo) e la versione per Netbook usava un nuovo ambiente desktop di nome **Unity**. Nella release 11.04, però, è stato adottato anche per la versione principale. Da tempo Ubuntu aveva dato segni di insoddisfazione con il desktop Gnome: il 3 non era considerato abbastanza stabile da essere incluso nella nuova release e Gnome 2 era già un pezzo d'antiquariato. In un certo senso Ubuntu non aveva scelta ma i cambiamenti non piacciono a nessuno e gli utenti hanno iniziato subito a lamentarsi del nuovo ambiente desktop. Ubuntu è comunque rimasto a fedele a Unity che oggi è molto migliorato ma un momento davvero critico si è verificato con la release 12.10 con cui gli utenti hanno notato che comparivano "suggerimenti" di Amazon quando inserivano dei parametri nel motore di ricerca su Internet. Neanche Gnome 3 è riuscito a scampare le controversie. Le critiche principali sono state tre. Prima di tutto, molti preferivano il modo di operare del vecchio Gnome 2 e non apprezzavano l'idea del cambiamento. In secondo luogo, gli avanzati effetti sul desktop richiedevano una scheda grafica ragionevolmente di qualità (e anche dei driver funzionanti). Si poteva attivare una modalità ridotta, ma andava a compromettere la fruibilità del desktop. Infine, sembrava progettato per l'uso sui cellulari o i tablet, ma persino oggi Linux sui dispositivi mobili (con l'eccezione di Android) non ha decollato, quindi sembrava inutile forzare gli utenti ad adottare questo modo di ragionare. Molti però hanno scoperto che, una volta abbandonata qualche vecchia abitudine e imparata qualche scorciatoia da tastiera (e possibilmente dopo aver installato **Gnome Tweak Tool**), il modo di lavorare di Gnome 3 può essere altrettanto efficiente, se non di più, di quello del suo predecessore. Gli utenti di KDE stavano a guardare, avendo già passato la fase di transizione del rinnovamento del desktop (anche se in quel caso era stato meno drastico di quello di Gnome) alla pubblicazione di KDE nel 2008. A questo punto dovremmo menzionare anche **Systemd**, ma non c'è molto da aggiungere a ciò che è già stato detto altrove. Il vecchio sistema **init** accusava i segni del tempo, ne è arrivato uno nuovo e migliore, non è piaciuto a tutti ma lo usiamo comunque e il popolo di Internet se la prende con Lennart Poettering. I giochi su Linux storicamente sono considerati un interesse di nicchia, e chi li usava fino a poco tempo fa lo faceva con **Wine**, che è stato pubblicato a metà degli anni Novanta. La situazione è cambiata nel 2013, quando Valve ha pubblicato il suo client **Steam for Linux**. Oggi ci sono più di 1.000 videogiochi disponibili per Linux, e sempre più ne vengono convertiti. Bisogna ammettere che molte delle versioni Linux di giochi di alto livello incorporano un layer **Wine** o un wrapper come **eOn** ma ci sono anche numerose realease indipendenti native per Linux. Valve ha persino fatto una versione OpenGL del gioco super splatter a tema zombie **Left 4 Dead 2** che ha avuto un successo maggiore della sua controparte per DirectX/Windows. Gli utenti Linux attualmente rappresentano circa l' 1% della base di pubblico di Steam ma la situazione potrebbe cambiare se avrà successo il piano di Valve di conquistare i salotti [e perché non i desktop!? – NdR] attraverso i box Steam, che sfruttano il sistema operativo basato su Debian Steam OS. La storia di Linux nel nuovo millennio, naturalmente, non finisce qui: negli ultimi due anni ci sono stati tantissimi sviluppi e anche momenti negativi, incluso il bug Heartbleed, oltre a una parziale risoluzione della lunga causa SCO-IBM e a una posizione molto meno aggressiva da parte di Microsoft. Lo spazio, però, è tiranno... LXP

> "Negli ultimi due anni ci sono stati tantissimi sviluppi e anche momenti negativi"



Salvare la storia del mondo

È possibile che un uomo, il suo piccolo team e un robot Open Source salvino la storia per immagini del mondo? Scopriamolo in questa intervista

Tom Smith si descrive come un inventore e imprenditore e ha creato Project Gado, uno scanner robotico Open Source che può digitalizzare grandi collezioni di artefatti storici. Spesso questi oggetti non sono solo inestimabili, ma anche incredibilmente delicati. Nel nostro incontro con Tom Smith abbiamo discusso della nascita di Project Gado e di come la filosofia Open Source su cui è basato possa aiutare il progetto a salvare l'intera storia per immagini del mondo. Ecco l'intervista in esclusiva per i lettori di Linux Pro.

Linux Pro: Ti presenti come un inventore e un imprenditore: come sei arrivato a questa carriera? Come hai iniziato?
Tom Smith: Mi sono un po' sempre visto in questi termini. Ho iniziato da bambino a smontare gli oggetti. Compravo ogni genere di dispositivo usato a basso prezzo per smontarlo e capire come funzionava. A quattro o cinque anni assemblavo computer e avevo strumenti come trapani e avvitatori ben prima dell'età in cui a un bambino si dovrebbe consentire di iniziare a usarli. Per tutta la vita sono stato affascinato dalla scoperta. Project Gado è nato come risposta a un problema molto specifico. All'inizio volevo solo affrontare una questione molto concreta e sentita usando le tecnologie che conoscevo e applicandoci una buona dose di inventiva. Quello è stato il punto di partenza ma il progetto è cresciuto molto oltre i suoi orizzonti iniziali.

LXP: Puoi spiegare ai nostri lettori in cosa consiste esattamente il tuo progetto?
TS: Ci sono archivi fotografici di musei, giornali e ogni sorta di organizzazione composti di centinaia, migliaia o persino milioni di foto storiche che possono risalire a un secolo fa o più ma sono tutte in formato analogico. Molte organizzazioni non hanno le risorse per trasferire in digitale le loro collezioni. Può succedere che un ente abbia un milione di foto utili e interessanti a cui nessuno può avere accesso. Devono essere in formato digitale perché la gente possa accedervi e usarle a scopi di ricerca e affinché possano diventare un valore monetizzabile in modo da poter supportare delle attività. L'ente però non ha le risorse economiche per far svolgere a qualcuno il lavoro di scansione dei materiali. Ci sono diverse tecnologie per la digitalizzazione automatica, come uno scanner con auto feed ma non si può applicare in questo caso, con foto che magari hanno 150 anni. Le fai passare in uno scanner e basta un minimo intoppo nel processo, come la carta che si blocca, per distruggere un inestimabile pezzo di storia. La tecnologia non offriva una soluzione efficace e l'unica strategia possibile sembrava quella di assumere delle persone per passare le loro giornate a scansire immagini su uno scanner orizzontale. Se non potevi permetterti questa operazione, il tuo archivio storico era destinato a perdersi. Non era accessibile a nessuno se non ai pochi che avevano la possibilità di venire fisicamente alla tua sede e passare del tempo a spulciare le collezioni. L'ho scoperto, come sempre mi capita, attraverso un'esperienza pratica, andando all'Afro American Newspaper a Baltimora. È il giornale Afro Americano che opera continuativamente da più tempo al mondo, una realtà unica. Sono attivi dal 1892 e nel corso di tutti questi anni hanno costruito una collezione di un milione e mezzo di foto storiche. È così grande che riempie intere stanze della loro sede di Baltimora. Nel corso dell'intera storia del giornale, però, sono riusciti a digitalizzare solo 5.000 foto. In realtà in quel momento stavo lavorando su un progetto legato alla tradizione orale, perché ho alle

spalle studi antropologici. Il progetto era per la John Hopkins University di Baltimora e il mio obiettivo era di reperire delle immagini per illustrare i racconti che avevo raccolto intervistando i membri di una comunità locale. Non riuscivo però a trovare nessuna foto in formato digitale. Pensavo che tutte le immagini della storia di questa comunità fossero andare perse, poi mi sono unito a un altro ricercatore per andare all'Afro American Newspaper, sono entrato nella loro sede e ho trovato un archivio di un 1,5 milioni di fotografie. Includeva migliaia di foto della comunità che stavo studiando e mi sono reso conto che documentava moltissimi eventi di grande importanza di cui non c'era nessun altro riferimento visivo. C'era però un problema: tutto era in formato analogico e di conseguenza non si poteva realmente condividere con il resto del mondo. D'altra parte il giornale non aveva le risorse per digitalizzare la sua collezione. Mi sono chiesto, così, se si potesse usare la tecnologia per creare un dispositivo in grado di scansire autonomamente questo tipo di foto storiche tanto importanti quanto delicate. Con un dispositivo del genere archivi come quello dell'Afro American Newspaper si sarebbero potuti condividere con il mondo e la prossima persona interessata alla comunità e alla sua storia avrebbe trovato la collezione di immagini pronta da usare. Questo è stato il motivo per cui ho progettato il primo robot Gado nel 2010.

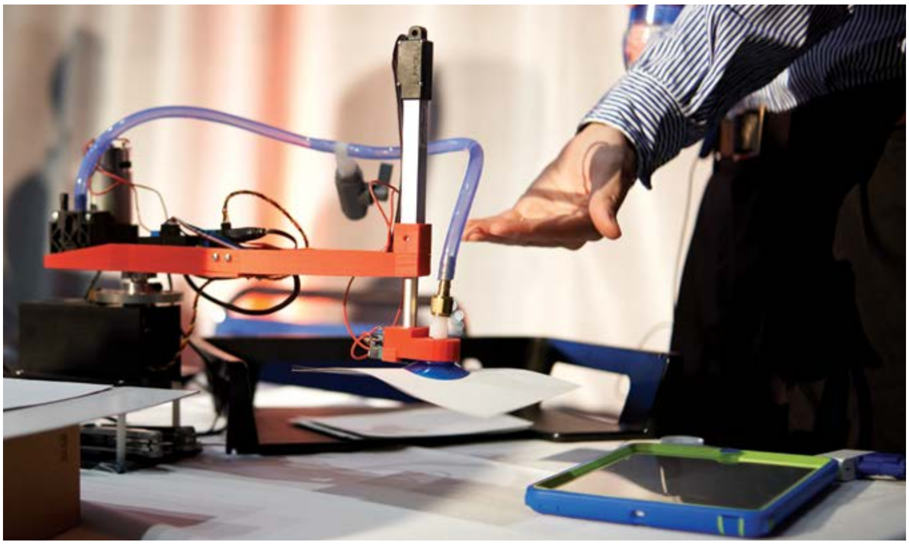
LXP: **Stavo proprio per chiederti di parlarmi del robot: deve essere delicato con questi artefatti insostituibili. Secondo te qual è il principale vantaggio di usare un robot come Gado per digitalizzare un archivio fotografico?**
TS: Secondo me il vantaggio principale è che permette all'archivista di sfruttare il tempo che avrebbe sprecato a digitalizzare i documenti manualmente per fare il lavoro davvero interessante e importante che solo una persona può svolgere. Quando abbiamo iniziato a lavorare sul progetto pensavamo che avremmo potuto sfruttare massicciamente la tecnologia per semplificare la vita dei gestori degli archivi e fare tutto il lavoro autonomamente. Cominciando in seguito a lavorare realmente con molti archivi e a provare praticamente la tecnologia sul campo, ci siamo resi conto che il principale vantaggio per l'archivista è che può limitarsi a far partire il processo, in modo che il robot possa occuparsi del tedioso compito di spostare le foto e fare le scansioni. Il robot non sostituisce la persona che opera nell'archivio, ma mentre lavora in background permette all'archivista di fare ricerche sulle origini della foto, fare lavoro di conservazione o collaborare con una persona in visita all'archivio. Di conseguenza secondo me il vantaggio maggiore che offre il robot è che toglie all'operatore umano un compito molto lungo da svolgere ma non altrettanto interessante e gli permette di focalizzarsi su compiti più stimolanti e piacevoli. Gli archivisti, in pratica, possono concentrarsi sulla parte del lavoro che li ha portati a scegliere la loro professione.

LXP: **Da quando è attivo Project Gado, quante immagini dell'archivio dell'Afro American Newspaper di Baltimora sono state convertite in formato digitale?**
TS: Abbiamo scansito 120.000 immagini della loro collezione. Quindi c'è ancora molto da fare, ci capita ancora di andare alla loro sede e trovare cartelle di documenti di cui non conoscevamo l'esistenza, ma abbiamo già digitalizzato un bel pezzo della raccolta di immagini e ci siamo concentrati particolarmente su alcuni degli elementi dell'archivio con la maggior rilevanza storica. Proprio in questi giorni abbiamo scoperto altre 30 foto del Dott. King, Martin Luther King Junior. Abbiamo anche ritrovato un'intera cartella di foto di Eleanor Roosevelt degli anni Quaranta. Con 120.000 foto già scansite c'era ancora un mondo da scoprire.

IL ROBOT GABO

"Nel corso dell'intera storia del giornale sono riusciti a digitalizzare solo 5.000 foto"

LXP: **Per creare il codice di Gado 2 hai usato Arduino e Python. Cosa ti ha portato a queste due scelte specifiche?**
TS: Ci sembravano coerenti con la storia dell'hardware e le motivazioni che ci hanno portato a svilupparlo, la sua "missione". Come ti dicevo, quando abbiamo iniziato a sviluppare l'hardware avevamo un obiettivo specifico in mente, la risoluzione di un problema attraverso un certo tipo di tecnologia e il primo robot che ho costruito è stato il Gado 1. Copriva tutto il tavolo e aveva una struttura in legno, quindi pesava quasi un quintale. Ci volevano tre persone per trasportarlo in un archivio. Ho impiegato circa tre mesi a costruirlo ed era un affare gigantesco che chiaramente non avrei mai riprodotto. Era sostanzialmente un prototipo per mettere alla prova la validità del concetto e lo abbiamo usato per scansire un migliaio di foto giusto per dimostrare che era possibile farlo. Quando avevamo annunciato il nostro intento, infatti, molti si erano dimostrati increduli e dicevano che non sarebbe stato possibile trovare una soluzione economica. Abbiamo invece dimostrato che si poteva creare un robot per la scansione in grado di acquisire fotografie storiche dal valore inestimabile senza danneggiarle con 500 dollari (circa 450 euro). Quando abbiamo deciso di sviluppare ulteriormente il progetto e passare al Gado 2 ci sembrava ovvio che la strada giusta fosse quella dell'Open Source. Così abbiamo progettato l'hardware in modo da rendere più facile possibile per altre persone analizzarlo e lavorare usandolo come punto di partenza. Da qui il passo a utilizzare Arduino e Python è stato breve: non solo sono ottime piattaforme e un eccellente linguaggio con cui lavorare, ma c'era già un'ampia comunità Open Source in continuo sviluppo. All'inizio tutto era scritto in Linux e i motivi di questa scelta erano stati esattamente gli stessi. È molto facile trovare persone che vogliono unirsi a un »



» progetto Linux e per loro è semplice attivarsi in tempi molto brevi. Ci sono inoltre tantissime funzioni integrate in Python, o Arduino o Linux, il che ti permette di non preoccuparti di aspetti già sviluppati: non devi scoprire da solo come controllare uno scanner con i driver TWAIN o come gestire una fotocamera DSLR. Per esempio, il robot scatta foto del retro di ogni immagine che acquisisce e questa operazione deve essere controllata dal computer. Farlo in Windows è un incubo: devi prendere i kit di sviluppo software di Canon e Nikon e integrarli nel tuo codice. In Linux tutto quello che ti serve è una linea di codice: **gphoto2**. È letteralmente così facile cominciare. Poi bisogna creare una community attorno all'hardware il più in fretta possibile.

LXP: Tu condividi tutto il codice su GitHub. Incoraggi altre persone a costruire i loro robot sulla base di Gado?

TS: Ah sì, assolutamente. Tutto ciò che ho fatto con il robot, il software di controllo e ogni altro dato è completamente Open Source. C'è tutto: non solo il codice che lo fa funzionare e il firmware Arduino, ma tutti i file per la stampa in 3D delle parti che servono. Anche questo è Open Source, come i materiali usati e i loro costi. Abbiamo persino creato alcuni kit e li abbiamo venduti a prezzo di costo ai primi utenti: li facciamo ancora per le persone che ci scrivono chiedendoceli. Se qualcuno vuole organizzare il suo gruppo di utenti possiamo fornire sostanzialmente tutto ciò che gli può servire. Non solo abbiamo voluto rendere ogni elemento Open Source ma siamo andati un passo oltre per rendere disponibile il materiale: ci attiviamo per portare altre persone a sperimentare con il progetto. Il messaggio non è solo: "Scaricatelo, è tutto gratis!" ma anche: "Ecco tutte le parti che vi servono, provate ad assemblarle, contattateci. Verremo a trovarvi, lavoreremo con voi e vi daremo una mano". La filosofia Open è stata implicita fin dall'inizio ed è il motivo per cui la tecnologia è progredita così rapidamente, al punto che in solo un paio d'anni siamo passati dal colosso che riempiva tutto il tavolo a un robot da 3 chili che può scansire 120.000 foto per meno di cinquecento euro.



LXP: Quindi il fatto di abbracciare appieno la filosofia Open Source ha aiutato a stimolare lo sviluppo di Gado?

TS: Sì assolutamente. L'importanza della comunità, che non è stato facile creare all'inizio, è uno degli aspetti che approfondisco più nel dettaglio nel corso delle mie presentazioni. Parlo di come è nata e cresciuta. Quando hai una comunità del genere puoi contare su uno dei grandi punti di forza dell'Open Source, ossia il fatto che il progetto diventa in grado di procedere e rafforzarsi autonomamente. Parlo spesso anche del nostro gruppo di utenti in Finlandia, che ha una bellissima storia. Avevamo annunciato pubblicamente che avremmo preparato dei kit. La stampa ne aveva parlato per il lancio e la gente cominciava a interessarsi. A un certo punto ci è arrivata una mail che diceva: "Vi scriviamo dall'università di Aalto a Helsinki, vorremmo uno dei vostri kit". Così ci siamo informati su come mandare in Finlandia il kit per la costruzione del robot senza avere problemi alla dogana e glielo abbiamo spedito. Tre mesi dopo ci hanno scritto che avevano assemblato il robot ma avevano delle domande da farci, così abbiamo iniziato a dialogare con la Finlandia. Abbiamo così scoperto che avevano organizzato un piccolo team di ricerca per lavorare sul robot. Avevano uno stagista che stava lavorando con loro per l'estate e alcuni ricercatori. Comunicavamo attraverso Skype, dovendo gestire una differenza di fuso orario di 11 ore, e mi alzavo alle 5 di mattina per collegarmi con loro e lavorare alla risoluzione dei problemi: siamo riusciti a renderli operativi con un robot funzionante. Poi non li abbiamo più sentiti per circa quattro mesi e alla fine abbiamo ricevuto un messaggio che diceva. "Abbiamo completamente riscritto il vostro codebase in Linux, potete darci accesso al vostro GitHub in modo per permetterci di caricarlo?". Naturalmente gli abbiamo dato l'accesso e abbiamo scoperto che avevano preso il nostro codebase Linux un po' raffazzonato e lo avevano completamente riscritto in modo da renderlo modulare e coerente con gli standard, per poi condividerlo in modo completamente gratuito. Sono passati circa altri sei mesi e abbiamo ricevuto un altro messaggio da loro che diceva: "Vogliamo fare un concorso di design industriale per portare gli studenti finlandesi di design a ristudiare la struttura fisica del robot e renderlo portatile in modo che si possa trasportare sul campo". Abbiamo risposto che per noi era fantastico, di procedere. Così sono finiti a fare un concorso con un paio di migliaia di euro di premi e moltissimi studenti di design finlandesi oltre a numerosi professionisti del design industriale si sono uniti per produrre disegni CAD completi, schemi e tutto ciò che poteva servire per realizzare un robot portatile da far venire nel musei. Hanno assegnato il premio e il vincitore ha costruito il suo progetto: una versione del nostro robot che si apre da un case che permette di trasportarlo come se fosse una valigetta. Tutto il progetto era Open Source: hanno condiviso tutto ciò che hanno fatto con il mondo e noi siamo rimasti ammiratissimi e abbiamo condiviso online e promosso il progetto. Poi hanno iniziato a portarlo nei vari musei della Finlandia e noi abbiamo cominciato a ricevere foto di bambini finlandesi che guardano il robot scansire le loro foto e artefatti personali. Alla fine hanno fatto tre giornate evento presso musei finnici e 300 persone sono venute a vedere il robot. Hanno anche fatto workshop con circa 30 persone per insegnar loro come usarlo per scansire archivi fotografici. Tutto questo è nato semplicemente dal fatto che abbiamo reso pubblico il materiale e loro hanno colto lo spunto. Secondo me è questo il grande potere dell'Open Source; puoi pubblicare il materiale, rendelo disponibile e cercare di spingerlo il più possibile ma una volta che la comunità è avviata nasceranno tanti progetti entusiasmanti che non ti saresti mai aspettato. È la forza che traina l'Open Source.

L'AIUTO DELL'UNIVERSITÀ DI AALTO

"Abbiamo riscritto il vostro codebase in Linux, potete darci accesso al vostro GitHub?"

LXP: Hai un'idea approssimativa di quanto possa essere estesa la comunità?

TS: Per quanto riguarda i gruppi con cui abbiamo lavorato direttamente, ne abbiamo uno in California, anzi due, uno alla periferia di Los Angeles alla Santa Anna Public Library, e ci siamo andati a fare workshop con gli studenti con la comunità. Abbiamo un gruppo

di utenti a Baltimora, quello in Finlandia, uno a Boston e poi ce ne sono diversi a cui abbiamo mandato dei kit con cui sappiamo che stanno lavorando ma non abbiamo collaborato direttamente in modo così attivo. Insomma, è in tutto il mondo. Si è diffuso davvero molto velocemente.

LXP: Potresti spiegarci come funziona Gado?
TS: Sostanzialmente si tratta di una serie di parti in plastica realizzate con la stampa 3D, tutte ottimizzate in modo che si possano stampare con un MakerBot, a casa o nel laboratorio di un maker. Il robot è sostanzialmente un braccio montato su un grosso servomeccanismo. A un'estremità del braccio c'è un attuatore che termina con una ventosa. Il robot essenzialmente ruota in un cerchio e all'inizio del cerchio c'è una pila di foto. Il robot ruota il braccio sopra la pila di foto, abbassa l'attuatore, attiva la ventosa, solleva una foto dal retro e ruota al centro del cerchio. Lì c'è uno scanner orizzontale, che può essere un qualsiasi modello commerciale (supportiamo tutti gli scanner usabili con Linux e abbiamo una serie di codici per far fare la stessa cosa a Windows). Il robot poi abbassa la foto fino allo scanner, attiva la scansione attraverso il suo software e digitalizza l'immagine alla risoluzione piena da archivio, che tipicamente nel nostro caso è di 600 DPI. Nomina il file della foto seguendo le convenzioni adottate dall'archivio su cui sta lavorando, per esempio usando i nomi delle collezioni o dei luoghi. Poi ruota di altri novanta gradi, disattiva la ventosa e rilascia la foto nel contenitore delle immagini già processate, infine ruota e torna al punto di partenza per ripetere l'intera operazione. Sostanzialmente è la stessa procedura che seguirebbe una persona, spostando la foto sullo scanner orizzontale e rimuovendola alla fine.
Un altro elemento che secondo me è importante menzionare è che sopra la pila di foto da scansire all'inizio del cerchio c'è una fotocamera DSLR o una Webcam, anch'essa attivata dal software, che fa una foto al retro di ogni immagine. Una delle sfide che stiamo affrontando ora è legata al fatto che, una volta scansite le foto, hai un'immagine raw di tutta la superficie dello scanner. La risoluzione è estremamente alta quindi ogni immagine avrà un file da 100 MB. Ognuno di questi verrà nominato secondo le convenzioni dell'archivio e alla fine ne avrai, per esempio, 120.000... qual è il passo successivo? Come si trasforma questo materiale grezzo in qualcosa che possa essere condiviso con la comunità o persino venduto su licenza per generare fondi per supportare le attività dell'archivio? Quello che stiamo iniziando a fare ora è considerare il materiale acquisito con le scansioni e cercare di usare del software per dare un ordine alle immagini, per cercare di creare metadati ed elaborare le foto in post produzione con la massima autonomia possibile. Lo scopo di fotografare il retro di ciascuno degli artefatti con la Webcam o le fotocamere DSLR è acquisire gli appunti che spesso hanno sul retro. Il fotografo può aver realizzato la stampa e aver annotato sul retro qual è il soggetto, oppure una data. Se appartiene a qualcuno potrebbe avere un marchio di copyright, che è un'informazione di cui bisogna essere a conoscenza. Ogni immagine è salvata con la foto del suo retro, per mantenere questi dati. Ora stiamo lavorando sul software per fargli svolgere mansioni come riconoscere i caratteri sul retro ed estrarre il testo, identificare i timbri, determinare automaticamente le date e passare tutte queste informazioni a un database, in modo che si possano fare ricerche sul materiale e aggiungere metadati in completa autonomia, con la stessa facilità con cui si eseguono le scansioni.

LXP: In quanto tempo pensate che possano essere pronte queste nuove funzioni?

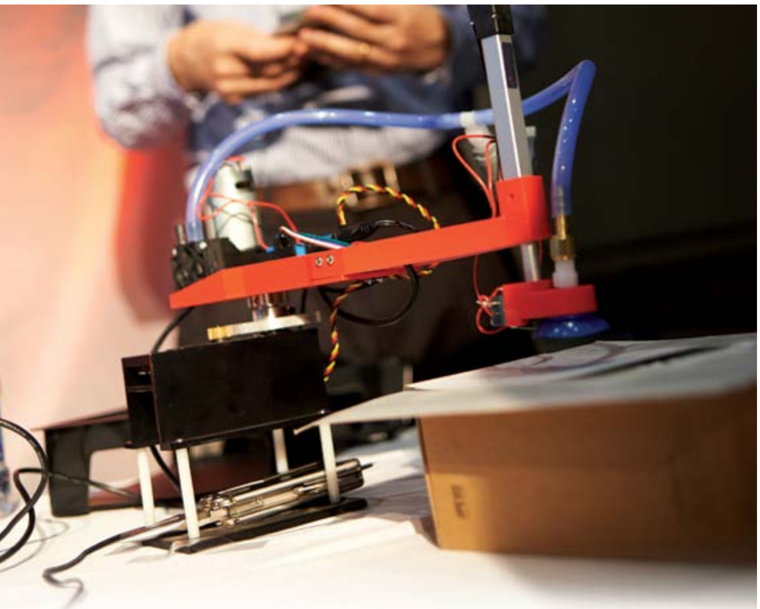




TS: Stiamo già testando delle versioni di prova con diverse organizzazioni. Uno dei nostri partner principali è Getty Images, l'azienda per la cessione di immagini su licenza più importante al mondo. Se vedi un'immagine storica in un telegiornale o su una rivista, probabilmente l'organizzazione che le ha distribuite è Getty Images. Lavoriamo con loro per portare le nostre foto in tutto il mondo e anche per consentire agli archivi di generare degli introiti in modo da poter finanziare ulteriori scansioni: è un meccanismo che si autofinanzia. Stiamo lavorando anche con varie altre organizzazioni che sfortunatamente non posso ancora menzionare, e stiamo facendo il beta testing di alcune delle altre collezioni che non abbiamo digitalizzato internamente.

LXP: Sarebbe possibile archiviare altro oltre alle fotografie?
TS: Forse. Quando abbiamo iniziato questo progetto originariamente volevamo raggiungere un obiettivo specifico per risolvere un problema esistente. La nostra mission originale era di creare un robot Open Source che potesse digitalizzare autonomamente archivi fotografici per meno di 500 dollari. Era quindi un obiettivo tecnico e specifico con uno scopo ben focalizzato e misurabile. Questo ha impegnato approssimativamente i primi due anni e mezzo del progetto. Una volta creata la community di utenti, ci siamo trovati a scansire centinaia di migliaia di foto con questa tecnologia e ci siamo resi conto che avevamo raggiunto il nostro obiettivo originario. La parte tecnica era risolta. Come procedere da quel punto? Cosa avrebbe potuto avere l'impatto »

» maggiore? Rispondendo a queste domande abbiamo deciso di ampliare la nostra mission alla digitalizzazione e la condivisione della storia del mondo per immagini. Al di là dello specifico obiettivo tecnico, ci siamo posti come meta di rendere la storia disponibile al mondo. Specificamente la storia per immagini, attraverso fotografie e artefatti grafici. Così, in ogni modo possibile, usando qualsiasi tecnologia utile, ovunque si trovino le risorse, che siano collezioni personali o archivi di musei, perseguiamo questo scopo. Nell'allargare i nostri orizzonti includendo tutto ciò che è visivo abbiamo iniziato ad abbracciare molte nuove tecnologie. Dalla domanda inziale: "Come si possono spostare e scansire automaticamente delle foto senza danneggiarle?", siamo passati alla successiva: "Come possiamo dare un ordinamento ai dati acquisiti?" per poi raggiungere molti altri quesiti, come la gestione delle diapositive, di negativi che magari hanno 100 anni, di materiale che si sta deteriorando, di libri stampati e rilegati. Stiamo cercando le risposte a tutto questo. Secondo noi il nostro impatto maggiore è il materiale visivo, specificamente fotografie e disegni. Certamente, però, vediamo nel nostro futuro la possibilità di scansire un'intera collezione... Ora che abbiamo ampliato la nostra mission al dì là dell'aspetto tecnico, ci guardiamo intorno e riflettiamo sul fatto che abbiamo creato una tecnologia, la abbiamo resa disponibile, abbiamo molte persone appassionate e competenti che ci lavorano ma si ripropone un problema simile a quello originario. Abbiamo creato uno strumento per l'uso da parte degli archivi, ma portarli a usare un robot Open Source sulle loro collezioni inestimabili o anche solo riuscire a integrarlo nel loro budget e trovare il personale per fare il lavoro pone un problema simile a quello iniziale. Abbiamo reso la tecnologia disponibile ma non è sufficiente. Dobbiamo promuoverla e lavorare nella pratica con queste organizzazioni, rendendo il tutto il più facile possibile, nello stesso modo in cui abbiamo creato un kit per rendere il più semplice possibile abbracciare il progetto per la comunità degli sviluppatori. Così recentemente ci siamo messi a lavorare direttamente con gli archivi con la stessa metodologia che avevamo adottato con l'Afro American Newspaper, passando dall'offrire semplicemente un robot che puoi costruire se vuoi a proporre una soluzione completa chiavi in mano con cui gli enti possono digitalizzare le loro collezioni. Grazie al fatto che abbiamo il robot e una relazione con Getty Images, possiamo fisicamente andare in un archivio e digitalizzare una quantità enorme del loro materiale gratuitamente, senza costi iniziali. Dato che siamo in grado di generare un introito dalla vendita su licenza dei materiali, possiamo permettere agli enti di pagare per il processo di digitalizzazione dopo che è stato completato, usando i proventi della collezione stessa. È facile immaginare le possibilità di questa metodologia di lavoro. Succedere spesso che qualcuno abbia una collezione ma non si possa permettere di portare del personale a lavorarci o non abbia le conoscenze tecniche per lavorare con il robot... ammesso che sappia della sua esistenza. Ora possiamo risolvere il problema: i proprietari delle collezioni possono venire da noi e lavoriamo con loro per la digitalizzazione senza che debbano pagare nulla in anticipo. Non solo fanno scansire e distribuire a livello internazionale il loro materiale, ma possono anche contribuire all'archivio storico e persino ottenere i fondi per procedere a ulteriori digitalizzazioni o perseguire i loro altri obiettivi.



LE SFIDE

"Non avevamo idea se fosse possibile o meno, ma abbiamo iniziato a lavorarci comunque"

Per noi questo è stato il nuovo passo avanti del progetto e un ottimo esempio è stata una collezione di foto digitalizzata di recente seguendo questo modello. Era una raccolta di immagini di un fotoreporter che ha vissuto e lavorato a San Francisco nei suoi 17 anni di carriera a un giornale di nome Sun Reporter. È venuto a mancare nel 1992 e ha lasciato la sua intera collezione a sua sorella. Erano venuti dei ricercatori a lavorare sulla collezione, ma non avevano i mezzi per digitalizzarla. Siamo stati in grado di intervenire e lavorare con la sorella del fotografo per digitalizzare i materiali e renderli disponibili.

LXP: Quante persone ci sono nel tuo team?
TS: Dipende da come li conti. A tempo pieno siamo in due ma se consideri le dimensioni della community che costruisce il robot siamo facilmente una cinquantina. Questo è un altro degli aspetti positivi dell'Open Source: possiamo creare grandi progetti perché abbiamo costi fissi molto bassi. Abbiamo una community di grandi appassionati, con ottime capacità e conoscenze, che supporta l'hardware e ci aiuta quando abbiamo un problema con un archivio. Se fosse un codice proprietario e chiuso, dovremmo avere un intero team di ricerca per risolvere i problemi, invece con l'Open Source lo possiamo condividere e l'intera community può lavorarci.

LXP: Vi preoccupa il fatto che qualcuno possa ottenere accesso ai contenuti insostituibili di un archivio, fare qualcosa di sbagliato e rovinare un artefatto? È un aspetto che crea apprensione? Spero che non sia ancora successo niente di simile!
TS: No, abbiamo fatto moltissimo lavoro senza mai danneggiare niente. Non abbiamo rovinato un singolo artefatto e neanche i nostri altri gruppi hanno avuto problemi di questo tipo... almeno non che sappiamo! Secondo me quando apri qualcosa alla comunità è straordinario quanto siano attente, rispettose e professionali le persone che ci lavorano. Non costruiscono il robot per andare a digitalizzare delle sciocchezze o per altri motivi futili. Nella nostra esperienza quando le persone si accollano il compito di creare un gruppo di utenti e iniziare a usare la tecnologia, di solito lo fanno esattamente con i nostri stessi intenti. Sono entusiasti all'idea di rendere la storia disponibile a tutti. Molti di loro operano all'interno di una grande organizzazione e se ottieni accesso agli archivi in prima istanza stai probabilmente lavorando con qualcuno che



è responsabile del materiale. Abbiamo visto molte collaborazioni. La Finlandia è stata un esempio perfetto in cui un'università ha sponsorizzato la ricerca che veniva svolta e sfruttata nei musei. Io però penso in termini ancora più ampi. È un rischio, ma è calcolato e vale la pena di correrlo. La nostra missione è digitalizzare l'intera storia per immagini del mondo… è molto ambiziosa.

LXP: La collaborazione internazionale sembra estremamente importante nel vostro progetto. Ci sono altri esempi di partner di altri Paesi che vi hanno contattato per il progetto Gado?
TS: Sì, assolutamente. Un ottimo esempio è un'azienda che si chiama Capture e ha sede nel Regno Unito. Abbiamo lavorato con loro sull'aggiunta di didascalie ad alcune delle immagini acquisite. Come dicevamo l'aggiungere metadati al materiale scansito è una sfida importante, così come trovare il modo per consentire al pubblico di fare ricerche all'interno degli archivi digitalizzati. Abbiamo lavorato con Capture su alcuni dei nostri documenti di maggior rilevanza storica. Hanno molta esperienza nel fare ricerche storiche sulle foto e capire chi vi compare, quando sono state scattate e altre informazioni di questo tipo. Abbiamo lavorato con loro su alcune delle collezioni per aggiungere questi metadati.
Gran parte del lavoro verrà svolto automaticamente con un nuovo sistema ma c'è ancora materiale su cui c'è la necessità di far intervenire un esperto. Partner come Capture ci mettono a disposizione questo tipo di expertise. Un'altra collaborazione che secondo noi è fantastica è quella con Getty Images per la distribuzione delle immagini: non solo sono presenti negli Stati Uniti ma anche in tutto il resto del mondo. Abbiamo iniziato a vedere le nostre immagini usate negli Stati Uniti: la rete televisiva di informazione CNN ha inziato a impiegarle e l'editore Scholastic ne ha inserite alcune in un libro di testo. Negli ultimi mesi però abbiamo visto una forte crescita dell'utilizzo internazionale delle nostre immagini. Le abbiamo viste dappertutto, dall'India al Giappone: le nostre foto sono passate sostanzialmente da un edificio a Baltimora a migliaia di chilometri di distanza e sono usate per illustrare libri di testo in Cina. L'elemento internazionale del nostro progetto, di cui la Finlandia è un ottimo esempio per quanto riguarda l'hardware ma di cui è importante anche l'elemento software, per la catalogazione e la distribuzione delle foto, si sta davvero espandendo in tutto il mondo. Anche nel digitalizzare un argomento specifico, la storia di una determinata comunità, volevamo che la nostra mission fosse internazionale. Anche se stavamo iniziando solo a Baltimora e San Francisco, stavamo pensando al mondo. E questi obiettivi si stanno traducendo in realtà.

LXP: Quali consigli daresti a chi aspira a diventare un inventore e trova difficoltà a far partire i suoi progetti?
TS: Consiglierei di tener presente che non è un processo istantaneo, è un percorso che inizi ma non sai dove ti porterà. Non sai nemmeno se funzionerà e troverai moltissime persone che non pensano che possa aver successo o andare in porto come vorresti che facesse. Ci saranno sempre moltissimi ostacoli e ragioni per non farlo. Ma ha sempre un senso, se credi in un'idea, cercare di perseguirla. Quando abbiamo iniziato c'erano tante domande senza risposta. Era un'idea un po' pazza costruire per meno di 500 dollari un robot in grado di digitalizzare inestimabili artefatti senza danneggiarli. Non avevamo idea se fosse possibile o meno, ma abbiamo iniziato a lavorarci comunque. Abbiamo avuto risultati e successi che non ci saremmo mai potuti immaginare. Credo che il miglior consiglio da parte mia sia di lanciarsi e provare. Fai un tentativo, vedi cosa succede. Sii pronto però a pensare che il progetto richieda tempi lunghi e che ci siano delle avversità. Preparati al fatto che ci saranno delle catastrofi lungo il percorso. Quando abbiamo iniziato a distribuire il nostro kit, c'era un problema con una delle parti: non funzionava. Così hanno iniziato a scriverci che non andava e noi abbiamo girato la questione alla comunità chiedendo cosa secondo loro avrebbe funzionato meglio. Abbiamo ottenuto moltissime risposte utili e interessanti, abbiamo progettato una parte completamente nuova per sostituire quella che non andava, l'abbiamo data alla comunità e come per magia il nuovo robot funzionava perfettamente! Quando si inizia un nuovo progetto bisogna aspettarsi delle sfide… ma nel processo che porta a risolverle si possono ottenere risultati che all'inizio non ci si sarebbe mai potuti immaginare. LXP



Premiata Amministeria Dott. Brown

Jolyon Brown
Quando non fa consulenze su Linux, Jolyon passa il suo tempo lavorando lavorando alla sua grande ambizione: trovare un motivo per usare Emacs...

Tecniche esoteriche per i sysadmin dai recessi più impenetrabili della sala server

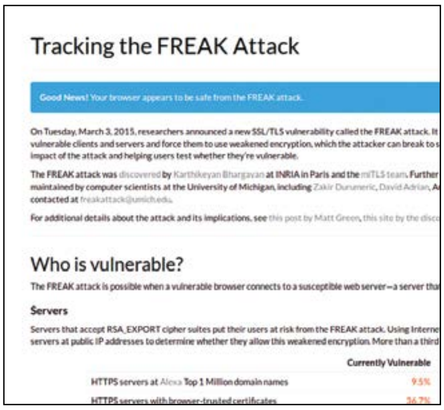
Fate subito un backup

Svolgo l'attività di amministratore di sistema da così tanto tempo che mi trovo spesso ad avere un paio di decenni di esperienza in più dei miei colleghi più giovani all'inizio della loro carriera (un fatto che turba sempre il ventenne che abita nella mia testa). Ogni tanto uno di questi ragazzini si toglie le cuffie, si avvicina, in sandali e felpa, alla mia scrivania e domanda al vecchio qualche perla di saggezza (o forse vuole solo assicurarsi che sono sveglio/vivo). Il consiglio numero uno che do a chiunque (che sia abbastanza paziente da attendere che trovi i miei occhiali e il cornetto acustico) è questo: avere sempre, ma proprio sempre, un backup. Anche se può sembrare ovvio, è una cosa spesso trascurata, di cui si parla in tono ironico ("che probabilità c'è che succeda?"), o deliberatamente ignorata per fare cose più prioritarie. Il consiglio non vale solo per le operazioni più importanti (come il ripristino di una copia di un database) ma anche a quelle più semplici e banali. Modificare un file? Avete sottomano una copia fatta in precedenza? Siete in grado di recuperarla dal sistema di controllo delle versioni? Qualsiasi compito stiate svolgendo, come amministratori di sistema dovete sapere come uscire da ogni situazione per tornare allo stato precedente. E, ugualmente importante, dovete verificare che il processo funzioni. Ora vi delizierò con un'esperienza personale. Una volta fui coinvolto in un'interruzione del servizio che durò tre settimane. La causa fu una serie di errori, eventi casuali e un allineamento tra i pianeti che ebbero come risultato una successione di turni di più di 24 ore, week end persi e un sacco di stress. I piani di emergenza e di backup fallirono e si scoprì che i contratti di assistenza non esistevano. L'iniziale scarica di adrenalina dovuta all'incidente svanisce in fretta in questo genere di situazioni. Meglio avere una vita tranquilla e accontentarsi di sapere che, nel caso dovesse succedere il peggio, voi e la vostra squadra siete in grado di uscirne per tornare a bere il vostro tè.

FREAK attack mette alla prova gli amministratori

Un'altra falla nelle librerie SSL ha causato una corsa all'aggiornamento dei sistemi

Lo scorso marzo, a seguito di numerose e importanti falle di sicurezza, sono state rivelate alcune vulnerabilità in parecchi pacchetti e librerie crittografiche open source. I gruppi di lavoro di Microsoft e Inria stavano cercando di esplorare un bug nella macchina a stati di queste implementazioni di TLS e hanno scoperto una serie di falle che erano presenti da anni. Una di queste è stata battezzata FREAK (**Factoring RSA Export Key**). Sfrutta le suite di cifre "da esportazione" che il governo degli Stati Uniti all'inizio degli anni '90 aveva obbligato a usare nelle versioni "internazionali" dei browser. All'epoca la crittografia era considerata dagli USA come un'arma ed era soggetta a ogni genere di restrizione nelle esportazioni. Di conseguenza il software crittografico doveva essere in grado di trattare queste suite con dimensione ridotta delle chiavi, finché le regolamentazioni vennero allentate e le dimensioni maggiori delle chiavi con cui siamo familiari oggi non divennero la norma. Sfortunatamente gli algoritmi indeboliti in grado di usare questo genere di crittografia sono rimasti all'interno di molte di queste librerie e, nonostante fossero disabilitati per default, i ricercatori sono riusciti a indurre il software a usarli. Questo significa che un malintenzionato può far uso di un attacco *man-in-the-middle* per indurre il server e il browser a scambiare dati cifrati con chiavi deboli, che possono essere decifrati in poche ore. La maggior parte dei browser più diffusi (a parte Firefox



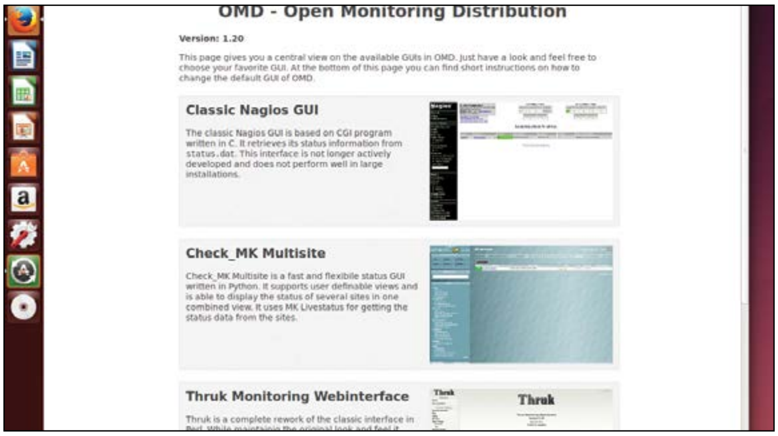
› **Il sito www.freakattack.com riporta comode statistiche sui siti Web affetti dal problema**

e Chrome a partire dalla versione 41) erano vulnerabili e anche molti siti (tra cui, ironicamente **nsa.gov**) lo erano. Il problema riguarda anche i client di posta elettronica e qualsiasi altro programma che per la crittografia faccia uso di una delle librerie vulnerabili. OpenSSL (la libreria più importante tra quelle coinvolte) ha silenziosamente rilasciato una patch a gennaio (CVE-2015-0204) per risolvere il problema. Anche se non se ne è parlato come l'anno scorso con Heartbleed (e non c'è stata una gara tra grafici per chi arrivava primo a creare un logo) l'attacco FREAK risulta comunque utile per ricordarci ancora una volta che un ciclo regolare di aggiornamenti del software è essenziale. Per questo in uno dei prossimi numeri mi occuperò di alcune delle strategie di aggiornamento per i sistemi Linux.

OMD: Open Monitoring Distribution

Avete bisogno con urgenza di una soluzione per il monitoraggio di tipo enterprise? Certo che ne avete bisogno. Niente paura, OMD è la soluzione

Il vecchio cavallo di battaglia delle soluzioni Open Source per il monitoraggio, **Nagios**, è in circolazione dal 1999. (All'inizio era noto come **NetSaint**, ma il nome è stato modificato in seguito a una diatriba legale.) Come è nella natura del movimento open source, nel corso degli anni sono nati fork e sono stati proposti miglioramenti e attualmente esistono numerose alternative (sia gratuite sia a pagamento). Per esempio c'è una versione ufficiale di Nagios (chiamata **XI**) che offre un contratto di supporto a pagamento. Esistono tuttavia moltissime installazioni della versione Open Source di Nagios, spesso nascoste in ambienti aziendali dietro schermate accuratamente progettate (usando strumenti come Nagvis), che hanno convinto parecchi arcigni responsabili IT che l'azienda sta utilizzando un rassicurante e costoso strumento di monitoraggio. Nagios ha ricevuto però numerose critiche nel corso degli anni. Le più frequenti riguardano la difficile scalabilità e il complicato processo di configurazione. Sono obiezioni che, in base alla mia esperienza, ritengo giustificate: i file di configurazione di Nagios sono fonte di confusione per i principianti e, a seconda del tipo e del numero di controlli applicati, un singolo server Nagios può effettivamente essere sottoposto a un carico elevato. I moltissimi plug-in disponibili per monitorare i più disparati tipi di sistemi e servizi sono poi sicuramente un vantaggio, ma vanno installati e configurati.



› OMD dispone di un certo numero di interfacce basate sul Web, dall'originale (leggi: antica) *GUI* di Nagios alla molto più piacevole *Multisite*

Nagios in confezione regalo

OMD (*Open Monitoring Distribution*) si pone l'obiettivo di risolvere questi inconvenienti, facilitando l'installazione e la configurazione e migliorando la scalabilità. Prende Nagios e lo impacchetta già preconfigurato e facile da installare. Oltre a questo nel pacchetto sono inseriti molti dei plug-in e degli extra più utilizzati, già pronti per l'uso al momento dell'installazione. Tra questi troviamo **Nagvis**, i Monitoring Plugin, Check_MK, Multisite, MK Livestatus, Mod_Gearman, rrdtool/rrdcached e altri ancora. In questo articolo mi concentrerò sull'agente **Check_MK** e sul front-end **Multisite**, ma voi potreste trovare gli altri più utili per le vostre esigenze, vi consiglio perciò di dargli un'occhiata. L'installazione del server centrale di OMD è davvero facile, dato che consiste in un unico pacchetto Deb o RPM che può essere scaricato da **http://omdistro.org**. Il sito Web si vanta di "fornire pacchetti per tutte le distribuzioni Linux di tipo enterprise", prima di aggiungere "...e anche per qualcun'altra, come Ubuntu". I soliti sospetti e i loro derivati sono coperti (RHEL, Debian e SUSE), mentre i sorgenti sono disponibili sia sul sito ufficiale (**http://bit.ly/OMDFiles**) che nel repository Git del progetto (**http://git.mathias-kettner.de/omd.git**). Il progetto dispone anche di un suo repository per i pacchetti ospitato all'URL **http://omdistro.org/start**, dove si trovano anche le istruzioni necessarie per installare il branch stabile o quello di test. Uno qualsiasi di questi metodi va bene per avere una installazione funzionante di OMD. Ultimamente i progetti in cui ho usato OMD avevano la tendenza a ospitare alcuni pacchetti, tra cui questo, localmente su un loro server, per ridurre le dipendenze da repository esterni. Questi pacchetti venivano poi installati dal software locale di orchestrazione: **Ansible**, **Puppet** o altri. Sul sistema CentOS (6.5) che sto usando per il test ho semplicemente scaricato **omd-1.20.rhel6.x86_64.rpm** (la release stabile al momento in cui scrivo l'articolo) in **/tmp** e l'ho installata con **rpm**:

```
rpm -i /tmp/omd-1.20.rhel6.x86_64.rpm
```

Il pacchetto viene installato in **/opt/omd**, che risulta essere un link simbolico a **/omd**. I manutentori del pacchetto hanno »

OMD: concetti e componenti

OMD è in realtà una collezione di estensioni già configurate per Nagios, con al centro Check_MK e costituisce un grande passo avanti rispetto al vecchio modo di fare le cose. L'agente esegue i controlli elencati nell'inventario sull'host su cui è installato, riconoscendo molti processi e servizi Linux senza bisogno di alcun intervento da parte dell'amministratore. Raccoglie il risultato dei controlli su tutto quanto gli è noto in modo che il sistema centrale di monitoraggio può limitarsi a contattare periodicamente ogni host, senza bisogno delle connessioni multiple che sono familiari a chiunque abbia lavorato con un'installazione di Nagios della vecchia scuola. Questi controlli "passivi" consentono a OMD di scalare molto più facilmente di una configurazione standard di Nagios.
C'è poi **Multisite** che è un'evoluzione della vecchia GUI di Nagios. Usa l'efficiente Livestatus per acquisire i dati sullo stato dei sistemi monitorati. Anche Nagios lo usa per le sue visualizzazioni. **WATO** è uno strumento basato sul Web per l'amministrazione e la configurazione del sistema. È molto potente e vale la pena di provarlo, così come conviene dare un'occhiata al modulo di Business Intelligence, che è in grado di modellare piattaforme più complesse di quelle che ho esaminato qui. Tutto questo consente un corretto monitoraggio "di servizio" dei vostri sistemi (cioè, abbiamo X server, quindi marchiamo un sistema come "critico" quando più di Y sono giù contemporaneamente), una delle cose più apprezzate dai gruppi di service management.

» scelto di installare ciascuna release in una sottodirectory della directory principale (per esempio **/omd/versions/1.20**), con un link simbolico che indica qual è quella *live*. Questo rende il passaggio da una versione all'altra estremamente facile: una cosa che gli amministratori di sistema apprezzano particolarmente! Viene anche registrato un nuovo servizio "omd" (visibile con il comando **chkconfig** su CentOS). OMD usa il concetto di sito, sostanzialmente una collezione di infrastrutture. Più siti possono coesistere uno accanto all'altro (diciamo, uno per ciascun ambiente). Io ho creato un unico sito chiamato **lxf**. Prima che procediate con la creazione del sito vale la pena di sottolineare che vengono definiti un nuovo utente e un nuovo gruppo con quel nome:

```
sudo omd create lxf
```

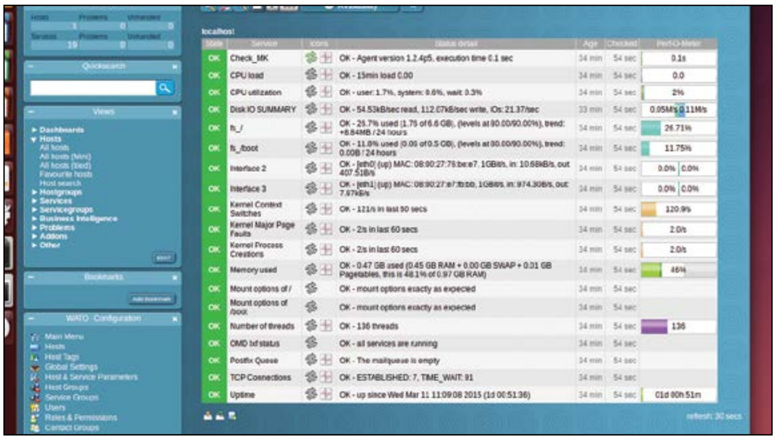
Se tutto funziona a dovere OMD vi fornirà qualche informazione sul sito, su come avviarlo e su come usare **su - lxf** per iniziare ad amministrarlo. I siti possono essere avviati dall'utente root o da proprietari del sito con il comando:

```
sudo omd start lxf
```

Il comando fa partire un certo numero di processi, tra cui un'istanza di http di **Apache** che ci permetterà di vedere l'interfaccia utente di OMD. Selinux potrebbe creare qualche problema: se l'avete abilitato impedirà (correttamente) a OMD di creare un host virtuale in ascolto sulla porta 5000. Questo può essere verificato con il comando **omd status lxf**, che mostrerà che Apache non è in esecuzione. Per le mie prove ho configurato Selinux in modalità permissiva (modificando **/etc/sysconfig/selinux**), ma questo potrebbe non essere accettabile nel vostro ambiente. Un'alternativa consiste nell'impostare il dominio http_t (in cui gira httpd) in modalità permissiva con il comando:

```
semanage permissive -a httpd_t
```

A questo punto, collegandosi con un browser all'URL **/lxf** sul server di test (**http://<server>/lxf**) si accede alla pagina di default di OMD. Vengono richiesti nome utente e password: **omdamin**/**omd** rispettivamente. A me interessa usare il front-end Check_MK Mulltisite: cliccando sul relativo link arrivo a una pagina molto carina ma... vuota.



› **L'agente *Check_MK* è in grado di scoprire da solo molti dei processi e dei servizi subito dopo l'installazione. Non occorre modificare i file di configurazione di Nagios**

Usare l'agente Check_MK

Non ha molto senso avere un sistema di monitoraggio con niente da controllare, quindi cerchiamo di fare qualcosa. OMD è dotato di un buon numero di agenti (script di shell) che vanno installati sui sistemi da monitorare. Si trovano sotto **/omd/versions/default/share/check_mk/agents** e ne esistono versioni per Linux e per la maggior parte degli Unix commerciali. Un'ulteriore sottodirectory contiene gli agenti per Windows. Quelli per Linux sono i più ricchi di funzionalità, ma ho usato con successo anche agenti per Solaris e Windows in ambiente di produzione. Piuttosto che copiare i singoli agenti ho sempre preferito usare la versione RPM (per una maggiore facilità di utilizzo con i sistemi di controllo della configurazione), ma recentemente sembra che siano stati spostati nell'area di download della versione a pagamento. Mi sembra una scelta un po' miope, ma evidentemente anche gli sviluppatori devono pagare le bollette. Un prerequisito per l'installazione di un agente su Linux è **xinetd**, quindi dovrete installarlo se non è già presente sul sistema. Un buon test iniziale di un'installazione consiste nell'installare un agente sullo stesso server di monitoraggio. La cosa è molto semplice da root, dalla directory **/omd/versions/default/share/check_mk/agents**:

```
# cp check_mk_agent.linux /usr/bin/check_mk_agent
# cp xinetd.conf /etc/xinetd.d/check_mk
# /etc/init.d/xinetd restart
```

Perché il monitoraggio abbia inizio occorre modificare il file **/etc/xinetd.d/check_mk**. Una riga al suo interno, riportata qui sotto, specifica quali server vengono monitorati dall'agente. Ho rimosso il cancelletto all'inizio della riga, lasciato l'indirizzo 127.0.0.1 e aggiunto quello del mio server al posto dei due indirizzi 10.0.X:

```
#only_from     = 127.0.0.1 10.0.20.1 10.0.20.2
```

Il file definisce anche la porta sulla quale l'agente risponde alle connessioni, il cui valore di default è 6556. Fatte queste modifiche posso verificare che tutto funzioni con il comando

```
telnet 127.0.0.1 6556
```

che produce un flusso di testo come output. Esaminandone il contenuto è possibile farsi un'idea dei vari controlli che l'agente esegue. A questo punto devo aggiungere l'agente al server di monitoraggio. Assunta l'identità dell'utente **lxf**,

Monitoraggio: è possibile un altro approccio?

Credo sinceramente che **OMD** (cioè **Nagios** e **Check_MK**) risolva per davvero i problemi di scalabilità e di difficoltà di utilizzo che hanno afflitto Nagios per un po'. Detto questo, esiste il pericolo che l'uso di questo genere di strumenti indichi che siamo prigionieri di una mentalità sorpassata per quanto riguarda il monitoraggio? Nagios ci fornisce una visione "su o giù" dei nostri sistemi e ci dice se si stanno comportando male avvisandoci quando escono al di fuori di certi parametri che noi abbiamo impostato. Esiste però un certo numero di sistemi proprietari che pretendono di andare oltre questo approccio semplicistico. Questi sistemi si vantano di rendere i gruppi di amministrazione più proattivi usando tecniche software più moderne di tipo *big data* e raccogliendo enormi quantità di dati di telemetria. Sono cioè in grado di trattare i problemi prima che diventino troppo gravi e aiutano a rintracciarne le cause, sempre con un minimo sforzo da parte degli amministratori. Tutto questo può essere descritto come *monitoraggio comportamentale* (per esempio registrando cose del tipo "il sistema è lento"): fare lo stesso con Nagios richiede un po' di riflessione e del tempo (ma non è impossibile se lo si usa insieme ad altri strumenti). Questo approccio è un primo passo nella direzioni di sistemi autonomi, in grado, entro certi limiti, di ripararsi da soli. Su quest'ultimo argomento c'è un libro: **In Search Of Certainty**, di Mark Burgess.

il proprietario del sito, modifico il file **etc/check_mk/main.mk** (notate come la directory si trovi al di sotto della directory home del proprietario del sito). Il file di default contiene solo poche righe. Cancellerò l'ultima riga e rimuoverò il cancelletto dalla riga che contiene "localhost":

```
# Put your host names here
# all_hosts = [ 'localhost' ]
all_hosts = [ ]
```

Il file principale di configurazione del sito è **main.mk**, che sostituisce i vecchi file di configurazione di Nagios. Questi esistono ancora (sotto **~/etc/nagios/conf.d**) e possono essere usati se richiesto, cosa che risulta comoda in caso di migrazione da una precedente installazione. Io preferisco usare **main.mk** nei miei progetti attuali e continuare a usare file di configurazione di solo testo piuttosto che WATO (vedere il box **OMD: concetti e componenti**). Questo perché preferisco conservare tutto in un sistema di configurazione con controllo delle versioni da cui estrarre i file da installare sul server di monitoraggio quando faccio una modifica. Rimanendo collegato come proprietario del sito **lxf** posso ora fare un inventario dei servizi da monitorare. Notate di nuovo il percorso al di sotto della directory home:

```
bin/cmk -I
bin/cmk -R
```

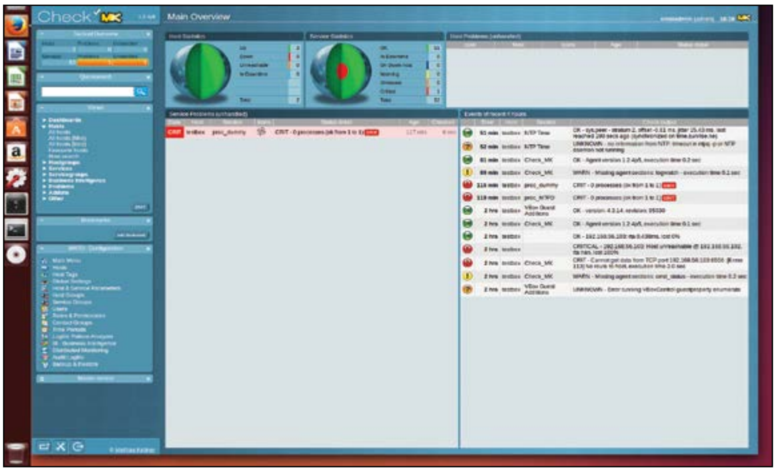
Se tutto è andato bene i comandi visualizzeranno una lista dei nuovi controlli che l'agente eseguirà e ricaricheranno la configurazione. Ritornato al browser, ricaricando la pagina vedrò i dati raccolti dall'agente sul server locale. Cliccando sul menu **Hosts**->**All hosts** a sinistra e sulla riga **localhost** che appare vedrò quello che check_mk sta monitorando. È in questo momento che si apprezza tutto il potenziale risparmio di tempo permesso da check_mk. L'inventario ha scoperto una serie di processi, li ha registrati insieme ai filesystem montati e ha iniziato a raccogliere dati su di essi e un'ampia gamma di metriche di sistema (CPU, memoria, connessioni di rete). Cliccando sulle icone a forma di grafico è possibile vedere i grafici prodotti da **pnp4nagios**. Il sistema conserva fino a un anno di dati sulle prestazioni in file RRD (*Round Robin Database*). Tutto questo in una frazione del tempo necessario alla configurazione di una installazione Nagios tradizionale. L'aggiunta di altre macchine avviene esattamente allo stesso modo dell'esempio con localhost. Installate l'agente, assicuratevi che il file di configurazione **check_mk** sotto **/etc/xinetd.d** contenga l'indirizzo IP corretto, aggiungetelo alla lista degli host in **main.mk** ed eseguite un aggiornamento dell'inventario prima di ricaricare la configurazione di monitoraggio. Ecco un esempio in cui ho aggiunto un altro host chiamato **testbox**:

```
# Put your host names here
all_hosts = [
'localhost',
'testbox|test',
]
ipaddresses = {
"testbox" : "192.168.56.103"
}
```

Dato che non ho un DNS nella mia rete locale ho dovuto aggiungere i dettagli dell'indirizzo IP del mio sistema di test in una nuova sezione del file (notare le parentesi graffe). Ho anche assegnato un "tag" a testbox. Si tratta di "test", separato dal nome dell'host da un simbolo di pipe. Un sistema può avere più di un tag, ognuno dei quali può essere usato per riferirsi a esso in un controllo. Io ho aggiunto le righe seguenti alla fine del mio **main.mk** prima di lanciare di nuovo **bin/cmk -R**:

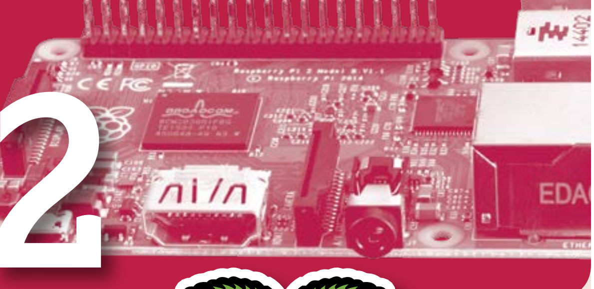
```
checks += [
# Tutti gli host con il tag "test" devono eseguire il demone dummy.
( ["test"], ALL_HOSTS, "ps", "dummy", ( "/usr/sbin/dummy", 1, 1, 1, 1 ) ),
]
```

Questo controllo usa uno dei moduli base di chek_mk chiamato **ps**, che esegue un semplice controllo sui processi in esecuzione. Ho deliberatamente chiesto di cercare un processo che non esiste per generare un "alert" critico (visibile qui sopra). Ci sono molte opzioni per questo modulo (vedere **http://bit.ly/ManCheck_mk_ps**), che è solo uno dei tanti che fanno parte per default del sistema. Il comando **bin/cmk -L** produce una lunga lista di quelli disponibili. L'installazione di OMD fornisce anche una serie di comodi plug-in (provate a dare un'occhiata sotto la directory degli agenti). Uno dei più utili è **mk_logwatch**. Una volta installato in **/usr/lib/check_mk_agent/plugins** sul nodo da monitorare andrà a cercare il file **logwatch.cfg** sotto **/etc/check_mk**. Nella directory degli agenti c'è un file di esempio. Copiate questi due file al loro posto, aggiornate l'inventario e poi ricaricate la configurazione di monitoraggio. Con la successiva esecuzione l'agente inizierà a estrarre dai log di sistema tutti gli alert che corrispondono alla sua configurazione, che è molto semplice da capire.
È molto veloce, è in grado di trattare file molto grossi e tiene traccia delle segnalazioni già emesse, quindi un riavvio dell'agente non provoca la notifica di vecchi errori già risolti. OMD è un progetto davvero ben fatto. Dato il poco spazio a disposizione ho illustrato solo i concetti base, ma sul sito degli autori (potete cominciare da **http://bit.ly/Check_mkIntro**) e nelle mailing list, molto attive e amichevoli, è disponibile una enorme quantità di informazioni (ed è probabilmente questa la cosa più impressionante di tutto il progetto). Grazie al fatto che su ciascun nodo i controlli sono eseguiti in un colpo solo e non singolarmente, sono riuscito a farlo crescere senza problemi fino a oltre 5.000 controlli in un ambiente di produzione di più di 100 nodi. Molte delle operazioni riguardano cose di cui non abbiamo parlato, come controlli SNMP su filrewall e switch di rete, sistemi Windows, controlli di tipo *business logic* e alcuni avanzati plug-in Nagios. Tornerò su OMD in futuro per esaminarne qualcuno. Per ora mi limito a raccomandarvi caldamente OMD per tutte le vostre esigenze di monitoraggio. LXP



› **La pagina di stato di *Multisite* è completa di grafici "nucleo di pianeta" che mostrano lo stato attuale dell'infrastruttura**

Pi & Pi 2



Tutte le news, le recensioni e i tutorial sul Raspberry Pi



CARRIE-ANNE PHILBIN
ci porta le ultime notizie dalla Raspberry Pi Foundation.

Benvenuti...

Il team didattico della Raspberry Pi Foundation, ci cui sono a capo, ama far conoscere al mondo gli insegnanti Raspberry Pi e per questo motivo la squadra è sempre in viaggio a fare conferenze, fiere ed eventi per mostrare le possibilità di Pi. L'anno scorso, abbiamo iniziato un programma didattico per gli insegnanti dal nome di Picademy e abbiamo invitato educatori da tutto il mondo al quartier generale di Pi per imparare divertentosi a sperimentare. Ha avuto un successo straordinario e abbiamo preparato circa 200 insegnanti nel nostro modo unico, nel corso di otto eventi. Il feedback è sempre stato positivo e dalla sua creazione nell'aprile 2014 il team ha acquisito rispettati membri della comunità per gestire sempre più eventi.

Picademy cresce

La Fondazione è molto ambiziosa: vogliamo formare centinaia di migliaia di insegnanti e finora questo non è stato possibile, malgrado il nostro fantastico team di super esperti. Grazie alla generosità e al supporto di Google.org, però, ora ci stiamo muovendo nella direzione giusta con il nuovo programma per educatori Picademy@Google che ha esordito a Leeds nel Regno Unito.
È un'opportunità per gli insegnanti delle elementari, medie e superiori per fare corsi di informatica e scienze su Raspberry Pi al Google Digital Garage, uno spazio al porto di Leeds. I corsi sono completamente gratuiti e il Garage sarà aperto fino a ottobre. Gli insegnanti possono iscriversi su **http://bit.ly/PicademyAtGoogle**.

Astro Pi è pronto al lancio

Raspberry Pi mira allo spazio ma prima gli serve un case speciale

Forse sapere che Raspberry Pi è stato invitato a fare esperimenti sulla Stazione Spaziale Internazionale (ISS) farà cambiare idea a chi ancora pensa che sia solo un fenomeno passeggero... Non si può però andare nello spazio con un Rasperry Pi qualsiasi in tasca... serve un case speciale e luccicante per l'occasione.
La Pi Foundation ha rivelato i dettagli e delle foto dell'esclusivo case in alluminio adatto al volo spaziale in cui verrà fissato il Raspberry Pi da mandare alla Stazione Spaziale Internazionale ed è sicuramente un kit di tutto rispetto. Dato che la convezione termica è assente nello spazio il Pi deve essere accuratamente raffreddato dal case, dato che le norme della ISS sanciscono che nessuna superficie può superare i 45 gradi. Il case è creato da un blocco solido di alluminio e potete trovare altre foto all'indirizzo **http://bit.ly/AstroPiCase**.



Arriva Modulo

Il sistema di espansione modulare per Raspberry Pi

Ecco la nuova generazione di schede di espansione per il Pi: **Modulo**. Questa essenziale soluzione finanziata con il crowd-funding offre una sere di pratici moduli elettronici a incastro che rendono facile e ordinato creare dispositivi elettronici basati su Raspberry Pi. Il sistema, basato su Arduino, ha moduli che spaziano dai display OLED ai joystick. Per saperne di più visitate **http://bit.ly/LXFmodulo**.



Pi B+ costa meno

Un taglio di prezzo per il modello originale B+

Con un milione di Pi 2 già venduti e l'impegno della Pi Foundation a supportare i modelli precedenti, non sorprende che il modello originale **Raspberry Pi B+**, in vendita fino a poco tempo fa allo stesso prezzo del Pi 2, veda calare il suo costo per gli utenti. Apparentemente, le ottimizzazioni del processo produttivo implementate per il Pi 2 hanno influenzato anche il prezzo del B+ permettendo di abbassarlo. Per ulteriori dettagli: **http://bit.ly/PiPriceCut**.



Pipsta

A cosa può servire una microstampante pensata per la Raspberry Pi? LXP investiga



› **Pipsta è davvero piccola, ma produce stampe di discreta qualità, seppur solo in bianco e nero**

In breve

» **Una piccola stampante termica per stampare velocemente con la RP. Pipsta è accompagnata da una libreria Python integrabile nei propri progetti e uno chassis da montare per proteggere la RP.**

Vedendo questo piccolo dispositivo che si collega alla Raspberry Pi, la curiosità di capire come funziona e quali siano i suoi impieghi ci è venuta subito. **Pipsta** è una stampante compatibile con tutti i modelli di RP e viene venduto in un kit che si monta in un'ora circa, non richiede l'uso del saldatore e contiene tutti i componenti elettronici necessari. Il dispositivo è composto da tre parti: la stampante, una scheda di controllo e un case in plastica. La stampante è un'unità termica e richiede un particolare tipo di rotoli di carta che reagisce al calore emesso dalla testina di stampa per produrre testi e immagini. Sotto l'unità di stampa si trova la scheda di controllo che si interfaccia con la RP tramite una porta mini-USB e il rispettivo cavo USB. Questa scheda ha un alimentatore che si collega nella parte frontale dell'unità. Il terzo e ultimo componente è il case che protegge la stampante ed è composto da sei elementi che si incastrano tra loro senza richiedere particolari strumenti per il montaggio. La Raspberry Pi si inserisce nella parte inferiore di questo case e si deve collegare un cavetto a una delle masse presenti nel GPIO così da fornire un collegamento a massa alla testina di stampa. Il motivo per cui Pipsta richiede un'alimentazione esterna è che il GPIO della RP non è in grado di fornire sufficiente potenza. Nonostante la RP sia inserita nel case della stampante, si ha accesso comunque alle porte HDMI, USB, Ethernet e alla sua alimentazione. Anche il GPIO è raggiungibile, anche se l'operazione è un po' complicata perché bisogna passare attraverso una feritoia nel case. La stessa cosa è vera per lo slot SD, ma fortunatamente la relativa parte del case si può togliere per consentire un accesso più agevole. Infine si può connettere all'apposito slot anche una Webcam facendo passare il cavo attraverso le aperture di USB ed Ethernet nella parte posteriore del case.

Setup non perfetto

Il produttore ha incluso nella confezione una guida all'installazione che copre ogni aspetto del processo ed è arricchita anche da alcune risorse online ospitate su Bitbucket. L'installazione del software ci è sembrata abbastanza facile, anche se abbiamo dovuto fare un paio di modifiche alla configurazione per disabilitare la stampante standard e consentire a ogni utente del sistema di usare Pipsta; questa operazione potrebbe non essere facile per gli utenti meno esperti. Dopo aver installato il gestore di pacchetti **pip** occorre installare le varie dipendenze di Pipsta, che servono a gestire la conversione delle immagini e a creare QR Code. Fatto ciò è possibile scaricare ed estrarre nella home il software della stampante, scritto in Python, e alcuni esempi. Al momento il tool per controllare Pipsta è scritto in Python 2.7, ma il team di sviluppo prevede un futuro porting per la versione 3. Per testare la stampante abbiamo subito messo all'opera il primo esempio fornito dal produttore, chiamato **Basic Print**. Questo programma esegue un test di stampa che mette su carta un messaggio dal tono classico, "Hello World from Pipsta", e il risultato ci è sembrato buono. Scorrendo poi tra gli altri esempi che abbiamo scaricato, ci è balzato all'occhio quello chiamato **Image Print**. Esso mostra come stampare file PNG in scala di grigio preservando un buon livello di dettaglio. Difatti siamo riusciti a riprodurre il logo della Raspberry Pi e anche una foto: con nostra sorpresa, la foto è venuta meglio del logo! Quindi, a cosa mai potrebbe servire Pipsta? Esistono al momento applicazioni per la stampa delle previsioni meteo, per i messaggi dei biscotti della fortuna e per la stampa dei tweet di Twitter. Per esempio potreste usarla per stampare dei badge al prossimo evento del vostro LUG. Come con ogni accessorio legato alla RP, l'unico limite è l'immaginazione. LXP

Caratteristiche



Facile da montare
Il case della Pipsta si monta con facilità e consente di usare tutte le porte della Raspberry Pi.



Tanti esempi
Il sito Web di Pipsta è ricco di file d'esempio scaricabili per imparare a usare le funzioni di stampa.

Linux Pro Giudizio

Pipsta

Sviluppatore: Able Systems
Web: www.pipsta.co.uk
Prezzo: 90,47 €

Caratteristiche	8/10
Prestazioni	7/10
Facilità d'uso	7/10
Qualità/Prezzo	7/10

» *Un buon prodotto per le scuole e i LUG che vogliono creare progetti legati a device programmabili.*

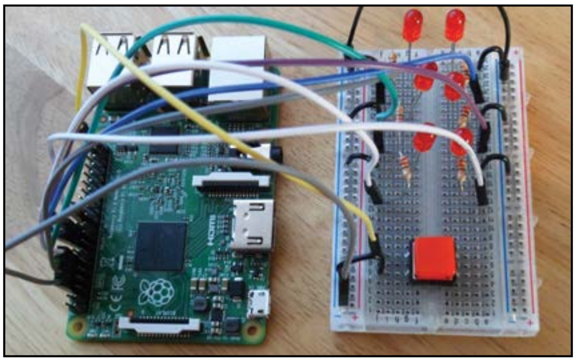
Il voto di Linux Pro 7

GPIO: simulare il lancio di un dado

La redazione di Linux Pro vi propone un'alternativa elettronica ai classici dadi a sei facce che si usano nei giochi da tavolo

Compatibile con tutte le RP

Gli umili dadi accompagnano da anni tutti i giocatori nei giochi da tavolo (e non solo). Ma come possiamo portare questo antico strumento nell'era elettronica del 21° secolo? In queste pagine realizzeremo una versione digitale di un dado usando la Raspberry Pi e qualche componente elettronico. Vi serve una qualunque versione della RP con Raspbian. L'elenco dei componenti elettronici prevede, invece, una breadboard, sei LED, sei resistori da 220 Ohm (per riconoscerli al volo i colori che li caratterizzano sono: rosso, rosso, marrone e oro), otto cavi elettrici femmina-maschio e sette fili maschio-maschio. Inoltre vi servirà del codice e uno schema circuitale che potrete scaricare dall'URL **http://bit.ly/LXP157_dadi**. Avete tutto? Ok, allora è il momento di collegare tutti i componenti alla RP.



› Il circuito sembra difficile? Niente paura: potete trovare il diagramma delle varie connessioni che dovete creare tra RP e breadboard all'URL http://bit.ly/LXP157_dadi

Setup del software

Il codice usato per questo progetto è scritto in Python 3; per digitarlo vi consigliamo l'uso dell'editor IDLE 3. Avviatelo usando **sudo** perché per utilizzare i pin del GPIO servono i privilegi di amministratore: per farlo, quindi, aprite **LXTerminal** e digitate il comando seguente:

```
$ sudo idle3 &
```

Con l'editor aperto potete iniziare a scrivere il codice. Cliccate su **File** -> **New** per aprire una nuova finestra e in essa selezionate la voce **File** -> **Save** per salvare il progetto. Questa è una buona pratica da seguire e limita la possibilità di perdere per distrazione il lavoro che si sta facendo. Il progetto inizia con l'importazione delle librerie necessarie. Le librerie sono dei "contenitori" esterni di codice che possono essere importati in ogni progetto allo scopo di sfruttarne le funzioni. Ne servono tre per questa guida: **RPi.GPIO** per abilitare l'accesso al pettine di pin GPIO, la funzione **sleep** della libreria **time** e la funzione **randint** presente in **random**.

```
import RPi.GPIO as GPIO
from time import sleep
from random import randint
```

Subito dopo bisogna definire come verranno usati i pin del GPIO. In questo caso userete l'impostazione stabilita da Broadcom, lo standard supportato dalla Raspberry Pi Foundation. La numerazione dei pin non è sequenziale, non c'è una progressione da 1 a 40 tra i pin, ma è la mappatura più popolare e più consistente.

```
GPIO.setmode(GPIO.BCM)
```

A questo punto si passa alla prossima sezione del codice che usa variabili contenenti il numero di pin del GPIO a cui sono collegati i LED e il pulsante. Le variabili servono a memorizzare dei valori che potrebbero cambiare nel corso del programma e semplificano la vita proprio nel momento della modifica.

Tip

Una breadboard è uno strumento che consente di costruire circuiti elettronici senza ricorrere al saldatore. Grazie a essa potete realizzare e testare velocemente un circuito per vedere se funziona in modo corretto. Esistono breadboard di dimensioni diverse: si va da dimensioni simili a quelle di un mattoncino grande del Lego a una board grande quasi come un foglio A4. Di solito costano pochi euro e sono spesso presenti nei tanti kit di elettronica che si trovano in Rete.

Attenti alla memoria

Avrete notato che in questo progetto abbiamo importato solo la funzione **randint** dalla libreria **random**, e non l'intera libreria. Perché? Semplice, importando solo le funzioni necessarie, il programma finale è più efficiente. Importando l'intera libreria random avremmo aumentato la quantità di memoria occupata. Inoltre anche la scrittura delle istruzioni sarebbe stata più complicata perché avremmo dovuto anteporre il nome della libreria a quello della funzione; per esempio, avremmo dovuto usare la funzione randint così:

```
random.randint()
```

mentre, importando solo la funzione che ci serviva, nel codice l'abbiamo richiamata con un più semplice

```
randint()
```

Insomma, nel primo esempio bisogna scrivere di più per avere lo stesso risultato. A dir la verità esiste un altro modo per importare una libreria intera senza però poi dover anteporne ogni volta il nome per le sue funzioni. Lo si può fare usando un carattere wildcard posizionando il comando d'importazione all'inizio del codice, così:

```
from random import *
```

Dal punto di vista della memoria questo metodo è meno efficiente rispetto a quello usato nel nostro progetto, ma non ha altre controindicazioni.

Per esempio, noi usiamo il pin 14 nel nostro progetto, ma se volessimo sostituirlo con il 15 e se non avessimo usato una variabile contenente il numero 14 (cioè se scrivessimo "14" a ogni riferimento al pin 14 invece del nome della variabile), dovremmo scorrere il codice riga per riga per cambiare il riferimento. Creando invece una variabile, identificata da un nome, a ogni sua ripetizione questo nome verrà sostituito dal valore in essa contenuta. Volendo cambiare il pin in uso, basterà modificare il valore assegnato alla variabile. Nelle semplici righe che trovate di seguito creiamo sei variabili, una per ogni LED, nominate da **d1** a **d6**; inoltre viene creata una variabile per memorizzare a quale pin è collegato il pulsante.

```
d1 = 14
d2 = 15
d3 = 18
d4 = 23
d5 = 24
d6 = 25
button = 21
```

Dichiarate queste variabili, dovete configurare gli input e gli output del progetto. Il metodo di input principale è il pulsante collegato al pin 21. Dovete impostare questo pin a HIGH (cioè acceso) usando il parametro **pull_up_down**. Poi bisogna indicare come output ognuno dei sei LED denominati d1-d6:

```
GPIO.setup(button,GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(d1,GPIO.OUT)
GPIO.setup(d2,GPIO.OUT)
GPIO.setup(d3,GPIO.OUT)
GPIO.setup(d4,GPIO.OUT)
GPIO.setup(d5,GPIO.OUT)
GPIO.setup(d6,GPIO.OUT)
```

Ora è necessario creare una funzione che simuli la rotazione del dado, cioè che calcoli il valore della sua faccia superiore:

```
def dice():
   for i in range(5):
      roll = randint(1,6)
   print(roll)
```

Si inizia assegnando un nome alla funzione: **dice** (dado in inglese). Potete vedere, poi, che le righe seguenti sono indentate. Questo è il modo usato da Python per indicare che il codice che segue appartiene alla funzione. Subito dopo c'è la parte che gestisce la rotazione virtuale del dado. Abbiamo usato un ciclo **for** che si ripete cinque volte (simulando la rotazione del dado). All'interno di questo loop abbiamo creato una variabile chiamata **roll** assegnandole un valore casuale compreso tra 1 e 6 (i valori presenti nelle sei facce di un dado). Alla fine del ciclo si stampa il numero scelto. La sezione di codice successiva (chiamata Razzle Dazzle) è un po' lunga e non la pubblichiamo qui, ma la trovate nel codice scaricabile dall'URL indicato all'inizio del tutorial. Il suo scopo è semplicemente quello di rendere un po' più divertente la simulazione del dado. Esso è composto da un ciclo che si ripete tre volte e che accende e spegne i LED pari e dispari. L'accensione di un LED (per esempio il primo) è gestita dall'istruzione **GPIO.output(d1,1)**; l'**1** rappresenta l'accensione. Una nota: al posto del valore 1 avremmo potuto scrivere TRUE. Per spegnere lo stesso LED bisogna sostituire l'1 con uno zero; in alternativa allo 0 si può usare FALSE.

```
   if roll == 1:
         GPIO.output(d1,1)
         sleep(3)
         GPIO.output(d1,0)
```

Positivo e negativo dei LED

I LED hanno due "gambe", quella più lunga è nota come anodo ed è la parte positiva (+); la gamba più corta, invece, chiamata catodo, è il negativo e va collegato alla massa (pin Ground o GND) della Raspberry Pi. Tra l'anodo dei LED e il rispettivo pin del GPIO abbiamo interposto un resistore da 200 Ohm, usato per ridurre la quantità di corrente in ingresso al LED così da proteggerlo e allungarne la vita. Il catodo di ogni LED si collega alla traccia collegata alla massa che arriva da uno dei pin GND della RP a scelta.

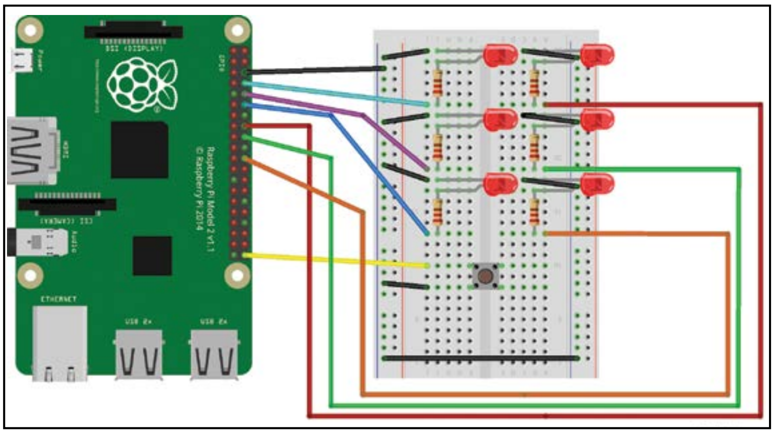
L'ultima parte della funzione **dice** è una serie di condizioni if... elseif che controlla cosa accade quando viene selezionato un certo numero (come prima, trovate online il codice completo):

```
   elif roll == 2:
         GPIO.output(d1,1)
         GPIO.output(d2,1)
         sleep(3)
         GPIO.output(d1,0)
         GPIO.output(d2,0)
```

In Python il costrutto else if è noto come elif. Provate a guardare la prima condizione: se il numero memorizzato nella variabile roll è 1 allora viene acceso il primo LED per tre secondi e poi viene spento. Se la prima condizione è falsa, invece (cioè roll è diverso da 1), viene controllata la condizione successiva e il processo si ripete fino a quando non viene trovata una corrispondenza. Per i valori compresi tra 2 e 6 si usa una serie di elif simili: cambia solo il numero di LED. Quando una di queste condizioni è vera, viene acceso il LED corrispondente. Infine, abbiamo creato un ciclo infinito per controllare quando viene premuto il pulsante. Premendolo, il suo stato passa da on (HIGH) a off (LOW) e ciò attiva la funzione dice() (cioè fa girare il dado simulato):

```
while True:
   if GPIO.input(button) == False:
      dice()
```

Assicuratevi di salvare il codice che avete scritto e poi selezionate la voce di menu **Run** -> **Run Module** per avviare il programma. Dopo una breve attesa, necessaria al suo caricamento, provate a premere il pulsante sulla breadboard. Dovreste vedere i LED illuminarsi per poi ritrovarvi per tre secondi un numero di LED accesi pari al valore casuale calcolato. Complimenti, avete appena creato il vostro primo dado virtuale! LXP



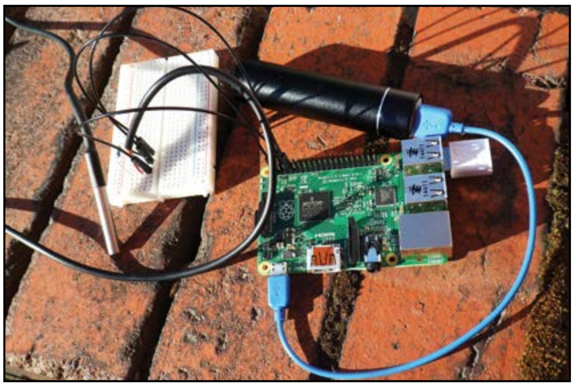
› **Noi abbiamo usato una breadboard con le piste ai lati, in questo modo si possono collegare tutte le masse dei componenti usando un solo pin di massa sulla RP**

Python: realizzare un monitor meteo

Trasformatevi in esperti delle previsioni meteorologiche e tenete sotto controllo la temperatura con la RP e Python

Compatibile CON TUTTE LE RP

La Raspberry Pi è un'ottima piattaforma per la realizzazione di esperimenti scientifici e potete usarla per acquisire ed elaborare con facilità dati provenienti dal mondo esterno... ma da dove si inizia? In questo progetto vi mostreremo come costruire un rilevatore di temperatura usando un sensore e confrontando i dati letti con quelli offerti da un servizio meteo online. Questi dati verranno poi disegnati su un grafico che mostrerà il trend di un dato periodo.
Ecco l'elenco delle cose che vi serviranno: una Raspberry Pi (qualunque modello) con una connessione Wi-Fi funzionante; un sensore di temperatura DS18B20, una breadboard, un resistore da 4,7kOhm, tre fili elettrici femmina-maschio.
Dal punto di vista software, userete le API fornite da **Open Weather Map** per leggere i dati meteo. Si tratta di API utilizzabili liberamente che offrono dati meteo molto dettagliati su ogni parte del mondo; oltre alle informazioni sul tempo, può anche mostrare previsioni o dati storici. Per usare queste API dovete registrare un account su **http://openweathermap.org/api** e ottenere una chiave. Ricordate di memorizzare questa chiave in un posto sicuro e che sono a uso personale. La userete per provare a recuperare i dati meteo della zona in cui vi trovate, ma prima è necessario installare il modulo Python **pyowm** che consente d'interfacciare uno script Python con il servizio online. Per installarlo dovete ricorrere a **Pip**, il Python package manager. Installatelo nella vostra Raspberry Pi se non lo è già e poi usatelo aprendo un terminale e digitando



› **Ecco il progetto finito. Potete alimentarlo usando una batteria aggiuntiva per smartphone così da piazzare il circuito in mezzo al giardino. Per proteggerlo dalle intemperie conviene però metterlo sotto un contenitore di plastica**

```
sudo pip-3.2 install pyowm
```

Ora avviate l'editor Python (IDLE3), aprite una nuova finestra andando in **File** -> **New** e iniziate il codice importando i moduli necessari. Pyowm è il wrapper Python per OpenWeatherMap, mentre dal modulo **time** dovete importare solo la funzione **sleep** che servirà successivamente per controllare la velocità di esecuzione del programma:

```
import pyowm
from time import sleep
```

Subito dopo bisogna definire una funzione che consente l'interrogazione del servizio meteo usando un solo comando invece di una lunga sequenza di istruzioni:

```
def get_weather(n):
```

Si inizia assegnando il nome **get_weather** alla funzione. Abbiamo inserito un argomento (**n**) tra le parentesi tonde; è un'opzione per cambiare il luogo di cui si cercano i dati meteo. Poi si crea una variabile **owm** che contiene la vostra chiave:

```
  owm = pyowm.OWM
 ('COPIATE QUI LA VOSTRA CHIAVE')
```

Dovete copiare la vostra chiave tra apici singoli per assicurarvi che Python capisca che si tratta di una stringa. Nota: questa linea è indentata per mostrare che appartiene alla funzione appena creata. Subito dopo serve un'altra variabile:

```
    observation = owm.weather_at_place((n))
```

Questa variabile si chiama **observation** e si usa come scorciatoia al nome di funzione che rappresenta la posizione usata nello script.

Controllare il tempo

Servono altre due variabili da usare come scorciatoie, una per recuperare i dati meteo e una per leggere la temperatura in gradi Celsius:

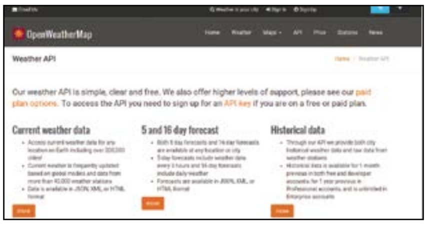
```
    w = observation.get_weather()
    a = (w.get_temperature('celsius'))
```

Nell'ultima linea della funzione usate la funzione **print** per stampare un messaggio per l'utente, qualcosa del tipo "La temperatura attuale a...". Per creare questa stringa adopererete la concatenazione per inserire il posto di cui avete letto i dati (rappresentato dalla variabile n) e, in fondo, il contenuto dell'altra variabile **a** contenente la temperatura. Quest'ultima variabile è piuttosto speciale perché in realtà si tratta di un dizionario Python che memorizza i valori usando una coppia chiave-valore. In questo caso indicherete a Python di cercare la chiave **temp** e di mostrare il suo valore:

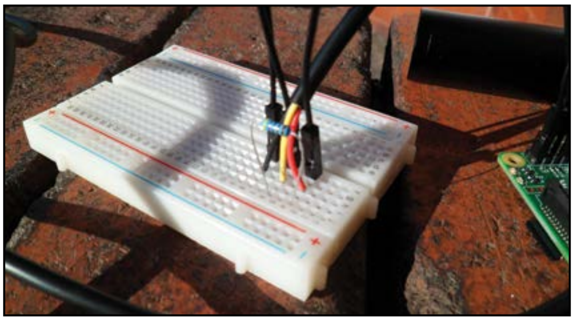
```
    print("The current temperature at",(n),"is",a['temp'])
```

Le API utilizzabili con Python

In questa guida facciamo ricorso a due API (*Application Program Interfaces*) necessarie a recuperare i dati da un servizio Web e per creare un grafico partendo da determinati valori. L'uso delle API sono il metodo più veloce per potenziare i propri script Python e a volte il loro uso può essere anche divertente. Pensate, per esempio, di sostituire le API di Plotly con quelle di Minecraft. Potreste decidere di modellare il mondo del gioco in base alla temperatura esterna: se dove vi trovate dovesse nevicare, potreste fare in modo che la stessa cosa accada durante il gioco. Oppure potreste disegnare il grafico della temperatura usando i mattoncini del mondo di Minecraft. Altre due API popolari che potrebbero tornarvi utili sono legate a Twitter e corrispondono ai moduli Python chiamati **Twython** e **Tweepy**. Con essi potreste integrare nello script la capacità di pubblicare online i valori della temperatura (o altri in base al vostro progetto). Altrettanto utile potrebbe essere l'API del servizio Web **www.smspi.co.uk**: essa vi consente di farvi inviare dallo script, via SMS, i dati registrati.



› Le API offrono un metodo rapido ed efficace per estendere le capacità di uno script. Sono molti i progetti Open Source a metterle a disposizione della comunità



› Noi abbiamo usato una breadboard per creare il prototipo del nostro progetto, ma se volete dargli un aspetto più professionale e "definitivo" potete usare una Perfboard (http://en.wikipedia.org/wiki/Perfboard) o disegnare un PCB

La funzione di test è terminata, quindi salvate il progetto e selezionate **Run** -> **Run Module** dalla barra dei menu di IDLE. La shell Python prenderà vita ma non accadrà nulla, almeno finché non fornirete una posizione richiamando l'apposita funzione:

```
get_weather("London,uk")
```

In alternativa potete usare anche lo ZIP code:

```
get_weather("W12 7LJ")
```

Dovreste vedere un output simile al seguente:

```
# The current temperature at London,uk is 22.06
```

Se tutto è andato come doveva, avete appena terminato la funzione che recupera il valore della temperatura dal servizio OpenWeatherMap. È il momento di passare al sensore di temperatura.

Prepara il sensore

Il sensore di temperatura gioca un ruolo fondamentale in questo progetto. Viene usato per leggere la temperatura del luogo in cui vi trovate così da poterla confrontare con i valori offerti da OpenWeatherMap. Il sensore che abbiamo usato è il Dallas DS18B20 che ha l'aspetto di un transistor. A esso vanno collegati tre fili: quello rosso va al pin dei 3,3V della RP; quello nero porta alla massa (GND); quello giallo è la linea seriale per i dati e va collegato al pin 4 della RP (seguendo la mappatura del GPIO di Broadcom). Per effettuare i collegamenti in tutta sicurezza è opportuno ricorrere a una breadboard e a un resistore da 4,7kOhm da inserire tra i 3,3 V e la linea seriale (guardate lo schema nella pagina seguente).

Collegati tutti i componenti, alimentate la Raspberry Pi e attendete l'avvio del desktop di Raspbian. Per usare il sensore dovete configurare la vostra RP, quindi aprite un terminale e digitate la riga seguente:

```
sudo leafpad /boot/config.txt
```

Si aprirà il file di configurazione di boot. Scendete in fondo al file e aggiungete la riga seguente:

```
dtoverlay=w1-gpio
```

Salvate e riavviare la RP. Provate a vedere se tutto funziona: sempre da LXTerminal digitate

```
$ sudo modprobe w1-gpio
$ sudo modprobe w1-therm
```

Questi due comandi caricano i moduli del kernel necessari per accedere al sensore: esso leggerà la temperatura attuale e la scriverà in un file; potete osservare il contenuto di questo file nel modo seguente:

```
$ cd /sys/bus/w1/devices
```

In questa directory dovete cercare una cartella che inizia con **28-**, quindi digitate

```
$ ls
```

per vedere il contenuto del folder. Individuata la cartella entrateci con **cd**

```
$ cd 28-
```

e poi stampate a schermo il contenuto del file **w1_slave**:

```
$ cat w1_slave
```

L'output di questo comando è piuttosto lungo, voi dovete cercare la riga che inizia con **t=**. Nel nostro caso il valore letto era 17.125 che, diviso per 1.000, ci dà 17,125°, il che dimostra che il sensore funziona correttamente. Sarà quindi possibile usarlo per memorizzare i valori di temperatura grazie al codice Python che vi mostriamo di seguito. Come al solito si inizia importando i moduli necessari al funzionamento del progetto. In questo caso occorre il modulo **OS** per consentire a Python di interagire con il sistema operativo sottostante. Il modulo **glob** serve a gestire i nomi dei percorsi all'interno del filesystem, mentre **time** viene utilizzata per controllare la velocità d'esecuzione del programma:

```
import os
import glob
import time
```

A questo punto si può usare la funzione **system** di OS per caricare due driver del kernel, uno per il GPIO e uno per il sensore:

```
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')
```

Tip

Nella parte circuitale di questo progetto abbiamo piazzato un resistore da 4,7kOhm tra il filo dei 3,3V e quello che trasporta i dati letti dal sensore. Questo resistore, detto di "pull-up", serve a rendere stabile il segnale e ad assicurarsi che sia compreso entro i valori previsti.

»

» Subito dopo bisogna creare tre variabili. La prima gestirà la posizione del sensore di temperatura all'interno del filesystem. La seconda usa glob per trovare la directory che contiene i dati del sensore. L'ultima viene usata per aggiungere il nome del file che contiene i valori registrati alla variabile precedente:

```
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'
```

Ora arrivano due linee di codice particolari. La prima crea una variabile globale chiamata **temp_c** che può essere usata all'interno e all'esterno di una funzione che creerete tra poco. La seconda, invece, genera una lista vuota. Le liste sono simili agli array usati in altri linguaggi e in essa memorizzerete i dati di temperatura:

```
global temp_c
sensor = []
```

Serve ora la funzione che leggerà la temperatura memorizzata nella variabile **device_file** creata prima. Aprirete questo file in sola lettura, leggete il suo contenuto e poi richiudetelo. Più avanti stamperete questi valori nella shell:

```
def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines
```

La funzione seguente gestisce la lettura della temperatura dal sensore e manipola il flusso di dati per eliminare le cose superflue. Come mostrato prima, bisogna dividere per 1.000 il dato registrato così da ottenere la corretta temperatura:

```
def read_temp():
    global temp_c
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()

    equals_pos = lines[1].find('t=')

    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c
```

Ora serve un ciclo da eseguire cinque volte: a ogni iterazione si stampa la temperatura letta dal sensore, si aggiunge questo numero alla lista sensor e poi la si stampa aspettando 3.600 secondi (un'ora) prima di ripetere il processo:

```
for i in range(5):
    print(read_temp())
    sensor.append(temp_c)
    print(sensor)
    time.sleep(3600)
```

Completato il codice, salvatelo ed eseguitelo usando **Run** -> **Run Module**. Sentitevi liberi di cambiare l'intervallo di tempo d'attesa tra un ciclo e l'altro.

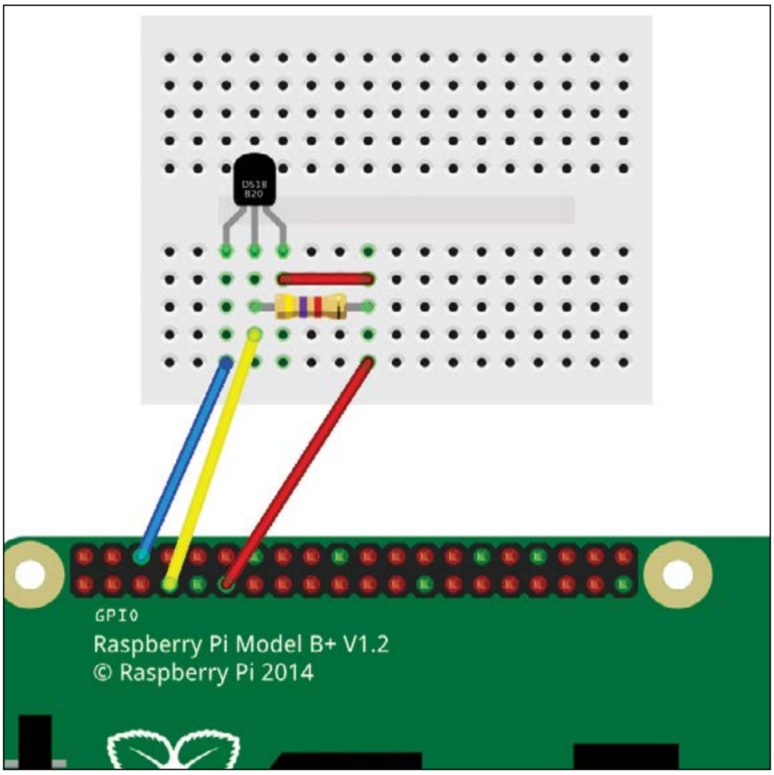
Impostare Plotly

Ora si passa alla parte grafica del progetto. Questa è gestita da **Plotly**, una soluzione grafica basata sul Web che fornisce un'eccellente API per Python. Dovete creare un account su **https://plot.ly** e ottenere anche in questo caso una chiave. Con questa sottomano, aprite LXTerminal sulla vostra Raspberry Pi e installate il modulo richiesto:

```
$ sudo pip-3.2 install plotly
```

A questo punto dovete indicare a Plotly il vostro nome utente e la vostra chiave delle API. Sempre da terminale digitate la riga seguente (è lunga, ma è un'unica riga):

```
$ python -c "import plotly; plotly.tools.set_credentials_
```



› **Potete scaricare il codice di questa guida dall'URL http://bit.ly/LXP157_meteo**

Acquistare il sensore di temperatura

Per questo tutorial abbiamo usato il sensore di temperatura DS18B20, che costa circa 5 € ed è un componente standard presente anche in alcuni kit di elettronica. A dire la verità ne esiste una versione modificata, con cavi più lunghi e resistente all'acqua, incluso nel kit **CamJam EduKit n.2** venduto dal sito **http://thepihut.com**. Purtroppo questo sito Web non vende nel nostro paese, per cui dovrete ricorrere al sensore standard, che va comunque benissimo. Sul mercato esistono ovviamente anche dei sensori alternativi, come i popolari DHT11 e DHT22. Questi sensori, però, usano un modo differente dal DS18B20 per interfacciarsi con la Raspberry Pi. I DHT11/22 utilizzano infatti un output analogico per inviare i dati alla scheda. Usandolo con Arduino non ci sarebbero problemi, visto che questa piattaforma è dotata di porte analogiche, ma la RP non ne ha. Quindi, se volete usare questo tipo di sensori con la RP, dovete procurarvi un ADC (*Analog to Digital Converter*) per consentire al piccolo computer di leggere i dati sulla temperatura. Questo ci porta a dire che il sensore da noi usato è la scelta più economica e anche quella più facile da mettere all'opera, ma voi siete liberi di agire diversamente.

```
file(username='IL VOSTRO NOME UTENTE', api_key=
'API KEY',stream_ids=['STREAM ID', 'STREAM ID])"
```

Trovate tutte queste informazioni all'URL **https://plot.ly/python/getting-started** dopo aver creato un account.

Assemblare il tutto

Finora avete creato le basi del progetto, adesso è il momento di unire le varie parti. Assicuratevi di avviare IDLE con i privilegi di amministratore usando il comando seguente da terminale:

```
$ sudo idle3
```

Il codice inizia con le istruzioni per importare i moduli necessari per il sistema operativo, i nomi dei file nel filesystem, le API per i dati meteo, la temporizzazione e i grafici:

```
import os
import glob
import pyowm
from time import sleep
import plotly.plotly as py
from plotly.graph_objs import *
```

Subito dopo serve un posto centralizzato in cui memorizzare le chiavi delle API di OpenWeatherMap e Plotly:

```
py.sign_in('USERNAME', 'API KEY')
key = ('API KEY')
```

Poi bisogna caricare i moduli del kernel per il sensore:

```
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')
```

A questo punto è necessario creare alcune variabili per memorizzare la posizione del sensore nel filesystem e del file in cui esso memorizza i dati. Servono poi tre liste. La lista **x** contiene il periodo di tempo considerato per la registrazione della temperatura, per esempio dalle 10 alle 14. La lista **y** è vuota ed è usata per memorizzare i dati della temperatura che arrivano da OpenWeatherMap, mentre **sensor**, anch'essa vuota, ospiterà i dati registrati dal sensore:

```
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'
global temp_c
x = ['10:00','11:00','12:00','13:00','14:00']
y = []
sensor = []
```

Dopo aver preparato lo spazio per registrare i dati bisogna creare le funzioni che li rilevino e li elaborino. La prima funzione legge i dati da OpenWeatherMap ed è molto simile a quella vista nella prima parte del progetto, con l'unica differenza che l'ultima riga di **get_weather(n)** viene usata per aggiungere i valori di temperatura alla lista y creata poco fa:

```
  y.append(a['temp'])
```

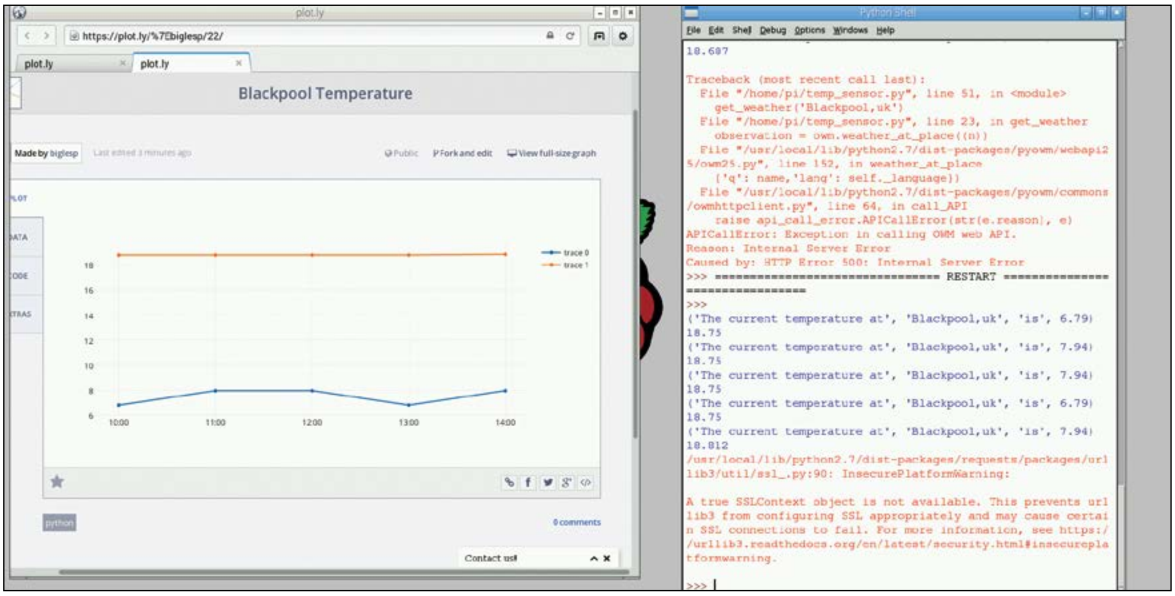
La seconda funzione, read_temp_raw() rimane la stessa dell'esempio precedente. A questo punto si passa al codice del loop da ripetere cinque volte. Esso recupera i dati meteo della località indicata e poi legge la temperatura registrata dal sensore. Entrambi i valori vengono scritti nelle rispettive liste, usando la funzione append. Giusto come controllo, a questo punto il codice stampa a schermo il valore letto dal sensore. Infine si mette in pausa il ciclo per 3.600 secondi (un'ora) prima di passare alla successiva iterazione del ciclo:

```
for i in range(5):
  get_weather('Blackpool,uk')
  read_temp()
  sensor.append(temp_c)
  print(temp_c)
  sleep(3600)
```

L'ultimo brano di codice da scrivere serve a disegnare il grafico con i dati. Verranno mostrati due tracciati: **Trace0** viene creata con i dati recuperati da OpenWeatherMap per l'asse y, mentre per l'asse x si usano i valori predefiniti legati all'ora di lettura. Per **Trace1** l'asse x è lo stesso, mentre y contiene i valori registrati dal sensore. Infine si impacchetta tutto in una lista e si inviano i dati a **https://plot.ly**, in questo modo:

```
trace0 = Scatter(x=(x),y=(y))
trace1 = Scatter(x=(x),y=(sensor))
data = Data([trace0, trace1])
unique_url = py.plot(data, filename = 'Blackpool Temperature')
```

Ora il codice è completo: eseguitelo come indicato prima. Al termine dell'esecuzione si aprirà in automatico una finestra del browser che mostra il grafico disegnato da Plotly. **LXP**



› **Quando eseguite il codice di questo progetto, esso invia il suo output alla console Python. Vedrete quindi i messaggi sulla temperatura attuale. Alla fine del loop verrà invece creato un grafico da Plotly che apparirà in una finestra del browser**

L'angolo di Android

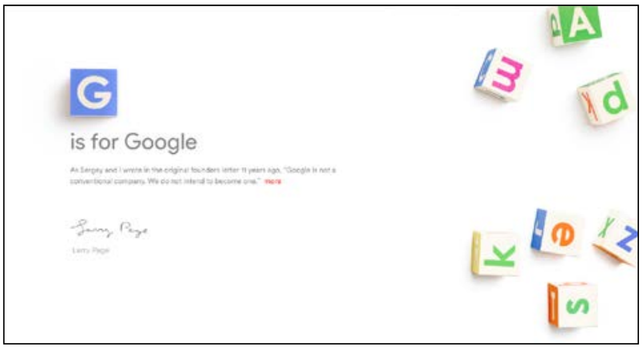
News, recensioni e guide sul sistema operativo libero per smartphone

Se hai news da segnalarci o dei commenti scrivici ad angolo_android@linuxpro.it

Nuova casa per Google

Arriva una nuova società che controllerà le iniziative del gigante di Mountain View

L'enorme numero di progetti è allo stesso tempo la forza di Google e una delle sue principali debolezze. Non si contano infatti le idee più o meno rivoluzionarie sviluppate negli ultimi anni dal gigante fondato da Larry Page e Sergey Brin: dalle auto senza guidatore ai Calico Labs, che studiano come prolungare la vita umana oltre la soglia dei 120 anni, al nuovo sistema Wing che prevede consegne postali con i droni, alla divisione Google X di cui si sa solo che studia progetti fantascientifici e top secret. Tanti progetti ma non sempre concretizzati, come si è visto con i Google Glass, che dopo anni e anni di sperimentazione sono rimasti un costoso gadget per pochi geek. Da un certo punto di vista si tratta delio stesso problema di frammentazione che affligge da sempre le varie versioni di Android, come dimostra il misero 15% di penetrazione ottenuto da Android 5.0 Lollipop a 10 mesi dalla presentazione. Per cercare di riorganizzare al meglio il tutto, e per dividere i progetti Web dagli altri, Larry Page e Sergey Brin hanno deciso di lasciare la guida della società da loro creata per dedicarsi ad **Alphabet**, una nuova azienda-madre di cui Google sarà solo una parte, anche se fondamentale. Il tutto attraverso una lettera pubblicata sul sito **abc.xyz**, in cui Page e Brin spiegano il loro progetto e all'interno della quale è nascosto un Easter Egg, cioè una delle 'sorpresine' che normalmente si trovano all'interno delle nuove versioni di Android e che questa volta rimanda al sito di Silicon Valley, la serie TV che è una parodia della stessa Google. Insomma, anche in questa occasione i fondatori di Google non rinunciano a prendersi in giro da soli. Il vero cuore (e la cassaforte) di Alphabet rimarrà però il motore di ricerca Google che verrà guidato da Sundar Pichai, l'ingegnere indiano fino a oggi Vice Presidente di Android, Chrome e App e che ora sarà responsabile anche di Maps, Photos e Google Now. Alla fine si tratta di una questione puramente economica, una volta messe al sicuro le attività più redditizie, gli eterni ragazzini Page e Brin potranno dedicarsi ai loro sogni adolescenziali senza rischiare di mettere a repentaglio il cuore della loro azienda da 500 miliardi di dollari. LXP



› **Il comunicato ufficiale di Page e Brin sulla nascita di *Alphabet* è naturalmente disponibile su Web: l'indirizzo non poteva che essere abc.xyz**

Android su Windows

Che Windows Phone sia destinato a una brutta fine lo hanno ormai capito in molti, inclusa Microsoft che infatti sta cercando in tutti i modi di rivitalizzare il sistema attraverso la compatibilità con le app Windows 10 e con quelle iOS e Android. **Project Astoria**, già disponibile in versione beta per le app iOS, entro l'autunno arriverà anche per Android. Si tratta di una mossa quasi obbligata per aumentare il numero di app disponibili per Windows Phone 10, ma che allo stesso tempo certifica una volta per tutte la resa del sistema operativo mobile di Microsoft, che in questo modo diventa quasi un'appendice di quelli di Google e Apple. LXP

OnePlus One

L'ultimo arrivato nel settore degli smartphone ha tutti i numeri giusti per dare filo da torcere anche ai migliori

I produttori cinesi stanno dimostrando di non avere nulla da invidiare ai loro colleghi giapponesi, coreani e taiwanesi che fino a oggi hanno dominato il settore degli smartphone. Ai marchi cinesi più rinomati come Huawei e Lenovo si è da poco aggiunto Xiaomi, che in Cina ha superato Apple e minaccia da vicino il numero uno Samsung. E ora arriva **OnePlus** che con il suo modello **One** si rivolge alla sempre più vasta categoria degli smanettoni che saranno sicuramente attirati dalle caratteristiche tecniche di eccellenza dalla Cyanogen Mod, la ROM custom più diffusa, unite a un prezzo senza concorrenti.
La struttura di OnePlus One è molto solida e di buona qualità, anche se per forza di cose sconta le dimensioni da 5,5 pollici del display e il peso di 162 grammi, con dimensioni più da tablet che da smartphone. La maneggevolezza è paragonabile a quella del Galaxy Note 3 di Samsung, ma con un bilanciamento migliore del peso, anche se rispetto al modello coreano manca il pennino e tutte le app collegate. La vera forza di OnePlus One sta nell'hardware che è a livello dei modelli top di gamma che costano il doppio: il SoC è il nuovo Snapdragon 801 nella versione più potente da 2,5 GHz con ben 3 GB di RAM, il chip LTE utilizza il protocollo da 150 Mbps in ricezione, il Wi-Fi è a doppia banda e il bluetooth utilizza il nuovissimo standard 4.1. Manca un'uscita video, ma oggi grazie alla chiave Chromecast da 35 euro è possibile duplicare il contenuto del display su qualsiasi TV con presa HDMI. Buona anche la resa dei due altoparlanti inferiori che sono stati realizzati in collaborazione con JBL.
Se proprio si vuole trovare un limite a OnePlus One, questo è la mancanza dell'espansione SD che costringe quindi ad accontentarsi dei 16 GB di memoria della versione base o dei 64 GB di quella superiore (che è consigliabile visto che costa solo 30 euro in più rispetto al modello base).

Maxi display

Il Full HD in uno schermo da 5,5 pollici non è più un record da quando LG ha presentato il suo G3 che presenta una risoluzione da 2.560x1.440 pixel, ben maggiore rispetto ai 1.920x1.080 pixel del OnePlus One. Questo però nulla toglie alla qualità dello schermo IPS dell'One che in più permette di regolare i vari livelli cromatici e la luminosità, ottenendo risultati completamente personalizzati. Volendo essere pignoli si potrebbe rilevare che il touchscreen non risponde all'utilizzo con i guanti, a differenza di alcuni modelli della concorrenza, ma si tratta francamente di un dettaglio poco influente. In generale la risposta al touchscreen è perfetta in ogni situazione e volendo è possibile anche usare il nuovo **Runtime Art** che velocizza notevolmente i tempi di avvio delle app dopo l'installazione. La fotocamera posteriore utilizza un sensore Sony Exmor da 13 megapixel con doppio illuminatore LED e obiettivo a sei lenti con diaframma f/2,0, in grado di registrare anche video in 4 K e in slow-motion a 720 p.
È anche presente uno stabilizzatore d'immagine hardware che riesce a compensare i movimenti più bruschi dell'utente. Si tratta insomma di una dotazione ai massimi livelli che soffre solo nella parte software, sottodimensionata rispetto alle possibilità hardware, un problema però facilmente risolvibile grazie alle tante app presenti nel Play Store.



› **Le plastiche adottate da *OnePlus One* sono di ottima qualità e sono disponibili molte cover intercambiabili anche in legno**

Android rinforzato

A prima vista la ROM c**ustom Cyanogen Mod** è identica alla versione di Android base utilizzata sui Nexus, in realtà però le differenze ci sono e riguardano principalmente la funzione Multi-windows e le personalizzazioni, nella barra delle notifiche e nella schermata di blocco. È anche possibile scaricare moltissimi temi originali che vanno a modificare le singole icone e i vari menu. I veri smanettoni potranno poi ottenere semplicemente i privilegi di root, visto che il bootloader è sbloccabile con un semplice comando da PC, anche se occorre ricordare che in questo modo verrà formattato completamente il sistema.
Nel momento in cui scriviamo il vero problema di OnePlus One è la scarsa disponibilità: per poterlo acquistare al prezzo di listino di 269 euro per la versione da 16 GB o di 299 euro per la versione da 64 GB è necessario avere l'invito di un amico che lo ha già acquistato oppure andarlo a cercare su uno dei tanti siti specializzati. Da settembre le cose dovrebbero però migliorare e comunque ci si può sempre rivolgere al mercato libero dove i prezzi stanno iniziando a scendere: su Amazon per esempio lo si può acquistare per 330 euro. LXP

LINUX PRO Giudizio

OnePlus One

Produttore: LG
Web: http://www.oneplus.net/
Prezzo: 269 € (16 GB), 299 € (64 GB)

Caratteristiche	8
Autonomia	8
Prestazioni	9
Qualità/prezzo	9.5

» *Un grande smartphone di fascia alta e prezzo basso con Cyanogen pre-installata.*

Il voto di Linux Pro 8.5



COD. LP145 — N 145 - SETTEMBRE 2014
COD. LP146 — N 146 - OTTOBRE 2014
COD. LP147 — N 147 - NOVEMBRE 2014
COD. LP148 — N 148 - DICEMBRE 2014
COD. LP149 — N 149 - GENNAIO 2015
COD. LP150 — N 150 - FEBBRAIO 2015
COD. LP151 — N 151 -MARZO 2015
COD. LP152 — N 152 - APRILE 2015
COD. LP153 — N 153 - MAGGIO 2015
COD. LP154 — N 154 - GIUGNO 2015
COD. LP155 — N 155 - LUGLIO 2015
COD. LP156 — N 156 - AGOSTO 2015

Se leggi Linux Pro, ti possono piacere anche...

OFFERTA SPECIALE

La guida completa ai droni: come sceglierli e come guadagnarci

COD. IMCS 2
€ 9,90



COD. RGV6
€ 9,90



COD. MICR1
€ 9,90



Completa la tua collezione ordinando gli arretrati a soli € 5,90 cad. su www.linuxpro.it/arretrati oppure utilizzando il modulo qui sotto

SCEGLI L'ARRETRATO CHE VUOI ORDINARE

SE VUOI ORDINARE VIA POSTA O VIA FAX, COMPILA QUESTO COUPON

Ritaglia o fotocopia il coupon, invialo in busta chiusa a: **Sprea S.p.A.** Via Torino, 51 20063 Cernusco s/n (MI), insieme a una copia della ricevuta di versamento o a un assegno. Oppure via fax al numero 02.700537672. Per ordinare in tempo reale i manuali collegati al nostro sito **www.linuxpro.it/arretrati.** Per ulteriori informazioni puoi scrivere a **arretrati@linuxpro.it** oppure telefonare allo 02/87158224 tutti i giorni dalle 14.00 alle 18.00

INSERISCI I CODICI delle pubblicazioni che desideri ricevere:

	€
	€
	€
	€
Totale Ordine	€

SCELGO IL SEGUENTE METODO DI SPEDIZIONE:
Indica con una ✗ la forma di spedizione desiderata

	Per **una rivista** spedizione tramite **posta tradizionale** al costo aggiuntivo di	**€ 3,90**
	Per **due o più riviste** spedizione tramite **Corriere Espresso** al costo aggiuntivo di	**€ 7,00**
	TOTALE COMPLESSIVO	€

Data |_|_|_|_|_|_| Firma del titolare____________________

NOME
COGNOME
VIA
N° C.A.P. PROV.
CITTÀ
TEL.
E-MAIL

SCELGO IL SEGUENTE METODO DI PAGAMENTO (Indica con una ✗ quello prescelto)

☐ Versamento su **CCP 99075871** intestato a **Sprea S.P.A. ABBONAMENTI Via Torino 51 20063 Cernusco Sul Naviglio MI** (Allegare ricevuta nella busta o al fax)

☐ Bonifico intestato a **Sprea S.P.A. Abbonamenti** sul conto **IBAN IT05 F076 0101 6000 0009 9075 871**

☐ **Carta di Credito** N.
(Per favore riportare il numero della Carta indicandone tutte le cifre)
Scad. CVV

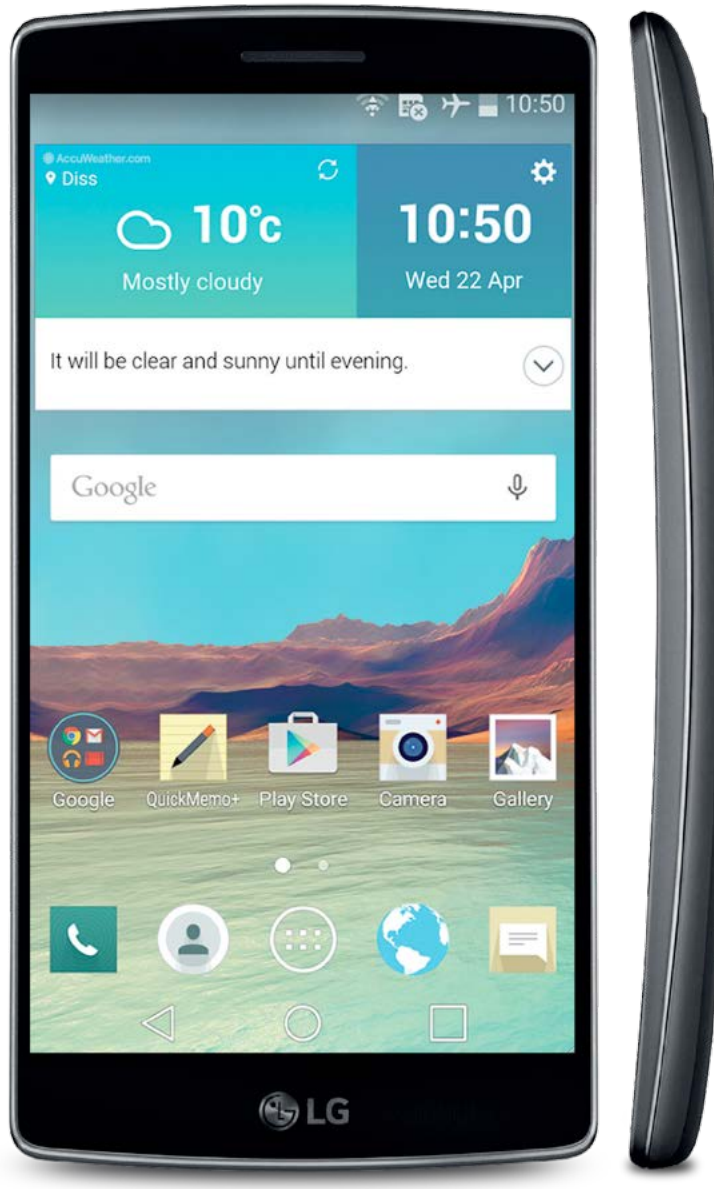
Nome e Cognome del Titolare della carta (può essere diverso dall'abbonato)

☐ CartaSi ☐ postepay ☐ MasterCard ☐ VISA ☐ Diners Club International ☐ American Express

LG G Flex 2

LG è tornata con una seconda versione dei suoi innovativi dispositivi dal design curvo, questa volta con specifiche davvero al top

LG sa decisamente fare schermi curvi. Ha fatto un buon lavoro con il primo G Flex e lo ribadisce con il suo successore, il **G Flex 2**. I più attenti noteranno che mentre Samsung ha scelto di curvare lo schermo da destra a sinistra, LG ritiene che il modo corretto di farlo sia dall'alto in basso. Lo schermo è un pannello da 5,5 pollici (più piccolo dell'originale, che ne misurava 6). Utilizza la tecnologia P-OLED, in cui il pannello in vetro è sostituito da uno strato di plastica, consentendo la caratteristica curvatura dello schermo. Lo schermo è brillante, con colori vibranti e allineato ai migliori in circolazione. Dietro allo schermo LG ha infilato tutta la moderna tecnologia che ci si aspetta di trovare in uno smartphone di questo prezzo: il più recente processore Snapdragon 810, 2 GB di RAM, espansione microSD, connettività LTE, una fotocamera da 13 megapixel con autofocus laser sul retro e una frontale da 2,1 megapixel, una batteria adeguatamente dimensionata e una cover posteriore rimovibile auto riparante, che utilizza una tecnologia migliorata rispetto alla precedente versione che aveva fatto il suo debutto sul primo G Flex. Il G Flex 2 è certamente attraente. Con il suo frontale brillante, circondato da bordi argento e i caratteristici pulsanti posteriori per spegnimento e volume, è possibile che riceviate commenti e domande dai passanti quando lo usate in pubblico. Quindi, l'hardware è bello, il software è bello... Ma come si comporta? Nel nostro test il processore Snapdragon 810 sembrava avere qualche problema. Quando si cominciano a sfruttare appieno le prestazioni del dispositivo, si scalda molto. Quando si surriscalda, la velocità del processore si abbassa, e la risposta diventa meno immediata. In definitiva, non si comporta nemmeno lontanamente come un telefono di punta del 2015. Quando state usando un videogioco, o persino durante l'installazione di applicazioni, il sistema inciampa e soffre di ritardi. Se siete davvero sfortunati, potrebbe addirittura ridurre la luminosità dello schermo, senza permettervi di rialzarla prima che il telefono si sia raffreddato sufficientemente. Questi problemi di prestazioni hanno anche un effetto negativo sulla durata della batteria. Abbiamo scoperto che l'autonomia del G Flex è molto inferiore a quella di dispositivi dalle caratteristiche simili, anche se dotati di batterie più piccole (abbiamo notato lo stesso problema su altri dispositivi con Snapdragon 810). Non è facile dire se questi problemi potranno essere risolti da LG con aggiornamenti software in futuro, o se ci sia un reale problema di fondo. È un vero peccato, perché il G Flex 2 è per tutti gli altri aspetti un dispositivo adorabile. Sfortunatamente, per come stanno le cose ora, non è un acquisto raccomandabile. LXP



› Il G Flex 2 ha uno schermo Gorilla Glass con pannello P-OLED, che utilizza un supporto plastico al posto del vetro. Come i fratelli G2, G3 e G4, il G Flex 2 ha pulsanti di accensione e volume sul dorso del telefono e questo ha permesso di avere un bordo più sottile

Scheda tecnica

» **Sistema operativo** Android 5.0.2 Lollipop
» **Processore** Snapdragon 810 octa-core
» **Dimensioni** 149,1 x 75,3 x 8,9 mm
» **Memoria** 16+2 GB di RAM
» **Peso** 152 grammi
» **Schermo** 5,5''
» **Risoluzione schermo** 1.920 x 1.080 pixel
» **Espansione** microSD

LINUX PRO Giudizio

LG G Flex 2

Produttore: LG
Web: www.lg.com\it
Prezzo: 699 €

Caratteristiche	8
Autonomia	6
Prestazioni	6
Qualità/prezzo	7

» *Ha ottime specifiche, ma i problemi di riscaldamento e rallentamenti sono gravi.*

Il voto di Linux Pro 7



ZTE Blade S6

Con l'S6 ZTE vuole rivivere le magie del Blade originale: esaminiamolo da vicino per capire se ci riesce...

Nel 2010 ZTE ha lanciato l'originale Blade. Aveva uno schermo WVGA da 3,5 pollici, all'epoca di tutto rispetto, e un processore Qualcomm Snapdragon. Il dispositivo ottenne un successo inaspettato, con specifiche che smentivano il suo prezzo conveniente e con il sostegno di una vivace comunità di modding. Giunti nel 2015 ci troviamo di fronte a un nuovo smartphone che potremmo definire l'erede spirituale di quel modello. Molti Blade sono andati e venuti, ma questo è probabilmente il primo ad avere quello speciale mix d'ingredienti che possono rendere un telefono di fascia media un modello top. Ciò che rende speciale l'**S6**, e che ha reso così l'originale è il rompere con la formula tradizionale adottata da molti suoi rivali. Processore non al top? No, il telefono integra un 64-bit, octa-core Qualcomm Snapdragon 615. Sistema operativo obsoleto con ampie modifiche? Al contrario, il Blade S6 è il primo dispositivo con processore 615 ad avere Android Lollipop. Poca RAM? Non particolarmente, visto che i 2 GB presenti su questo telefono sono lo standard del momento. Poca memoria? Non tanto, sono gli stessi 16 GB del Sony Xperia Z3 (che costa tre volte tanto) e può essere espansa tramite microSD. L'elenco potrebbe continuare. Aggiungiamo subito il Wi-Fi dual band, un'ampia gamma di sensori, un impressionante schermo da 5 pollici, un sensore che troviamo sulle fotocamera Sony di fascia alta, e si capisce subito come l'S6 abbia, per il suo costo, tantissimo da offrire. Naturalmente non ci sono specifiche di punta in ogni suo componente, ma da qualche parte è stato necessario fare dei sacrifici. Lo vediamo principalmente nel design e, in misura minore, nei materiali. Il dispositivo è un tributo palese all'iPhone 6, persino nel "Designed in California" marchiato sul retro della cassa. Questo non vuol dire che il look non sia adeguato. S'impugna bene, proprio come l'iPhone e, anche se completamente in plastica, è di buona qualità, con una minima flessione e delle accattivanti finiture in color argento sul dorso. Senza la necessità di dover isolare l'antenna, il design sembra meno esigente di quanto non sia sul dispositivo Apple. I 7,7 millimetri di spessore permettono di definirlo sottile e, vista la presenza di una batteria da 2,400 mAh, ciò sorprende. Anteriormente è di colore bianco, con tre pulsanti capacitivi blu retroilluminato nell'area inferiore che svolgono le funzioni back, home e menu (una pressione prolungata sul pulsante menu apre la pagina delle app recenti). L'area è dominata dallo schermo da 5 pollici a 720p con tecnologia touch In-cell. È ottimo e molto sensibile. Con il processore veloce e un sistema operativo reattivo e fluido (Lollipop con il suo runtime ART), garantisce ottime esperienze d'uso. ZTEpunta molto anche sulla fotocamera del Blade, mettendo in evidenza come il telefono integri una Sony IMX214, che troviamo anche negli Xiaomi MI4, OnePlus One e in altri telefoni economici con elevate specifiche. Secondo ZTE, forse troppo ottimisti, garantisce una qualità da reflex. Comunque, cattura ottime foto, aiutata da un'app fotocamera che può essere usata in modalità automatica o in una configurazione per esperti che offre un'ampia gamma di comandi manuali. Anche la fotocamera anteriore è ottima, con funzioni automatiche per migliorare i ritratti e per scattare solo quando il soggetto sorride. L'S6 ha anche quelle che ZTE chiama "sonorità Hi-Fi". In realtà, il dispositivo ha un solo diffusore collocato sul retro (con dei piccoli piedini che lo tengono sollevato quando posto su una superficie piana) e, anche se l'audio è di buona qualità, non è particolarmente elevato e nessuno è mai riuscito a riprodurre un suono stereo da un singolo altoparlante. LXP

Scheda tecnica

» **Sistema operativo** Android 5.0
» **Processore** Snapdragon 615 octa-core
» **Dimensioni** 144 x 70,7 x 7,7 mm
» **Memoria** 16+2 GB di RAM
» **Peso** 134 grammi
» **Schermo** 5"
» **Risoluzione schermo** 1.280x720 pixel
» **Espansione** microSD

LINUX PRO Giudizio

ZTE Blade S6

Produttore: ZTE
Web: www.ztedevice.it
Prezzo: 250 €

Caratteristiche	10
Autonomia	8
Prestazioni	9
Qualità/prezzo	8

» *Ottimo prezzo, peccato solo per la bassa risoluzione dello schermo.*

Il voto di Linux Pro 9



Recensioni

Tutte le novità in campo software e hardware testate e valutate ogni mese dai nostri laboratori

Se vuoi segnalarci qualche novità scrivi a recensioni@linuxpro.it

Una breve legenda

Ogni test di questa sezione è accompagnato da un giudizio che riassume con quattro indici numerici le principali qualità dell'applicazione o del prodotto hardware messo alla prova. I laboratori di Linux Pro assegnano un voto da 1 a 10 alle seguenti categorie:

Caratteristiche: fornisce tutte le funzioni di cui abbiamo bisogno? È innovativo?
Prestazioni: esegue in maniera efficiente le sue funzioni? È veloce e affidabile?
Facilità d'uso: dispone di un'interfaccia grafica chiara e facilmente fruibile? La documentazione che lo accompagna è sufficientemente completa ed esaustiva?
Qualità/prezzo: ha un prezzo competitivo? Vale i soldi richiesti per il suo acquisto?

Il nostro giudizio viene poi riassunto da un voto finale, espresso anche graficamente. Ecco la legenda dei voti:

10 Nulla da eccepire. Un prodotto praticamente perfetto.
8-9 Un buon prodotto. I pochi difetti presenti non sono gravi.
6-7 Compie il suo lavoro ma necessita di ulteriori sviluppi.
5-4 Deve migliorare prima di raggiungere un voto sufficiente.
1-3 Un completo disastro. Gli sviluppatori devono tornare alla fase di progettazione.

Ricordiamo infine che i software citati nelle sezioni Confronto e Da non perdere sono spesso presenti nel DVD sotto la voce "Rivista" sotto forma di codice sorgente o binario.

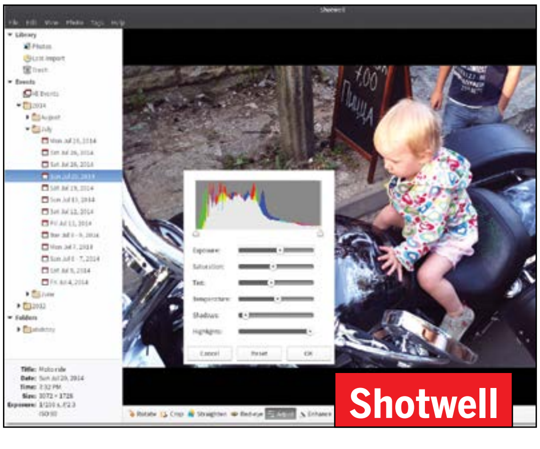
QUESTO MESE...



Chromebook Pixel (2015)



Pidgin



Shotwell

Chromebook Pixel

Il modello **Pixel** del 2015 di Google è il più potente Chromebook del pianeta ma ci domandiamo: vale la pena pagarlo così tanto?

Specifiche

» **CPU:** 2,2 a GHz Intel Core i5-5200U
» **GPU:** Intel HD Graphics 5500
» **RAM:** 8 GB di DDR3
» **Schermo:** Touchscreen IPS da 12,85 pollici 2.560 x 1.700
» **Archiviazione:** 32 GB di SSD
» **Porte:** 2 USB-C, 2 USB 2.0, uscita cuffie/microfono, lettore schede SD
» **Connessioni:** Intel Dual band Wireless-AC 7260; Bluetooth 4.0 LE
» **Camera:** 720 p HD
» **Peso:** 1,5 kg
» **Dimensioni:** 297x224x 15 mm (l x p x h)

Il Chromebook Pixel originale si è guadagnato il poco invidiabile titolo di Chromebook più costoso e ultra-potente in assoluto, con il suo schermo ad alta risoluzione e dall'insolito formato 3:2. Per il nuovo Pixel, Google ha scelto il medesimo schermo 2.560 x 1.700 e un chip ancora più potente, l'Intel Core i5 Broadwell; l'autonomia dichiarata della batteria è di 12 ore. E tutto questo a un prezzo inferiore. Sotto molti aspetti, il nuovo Pixel è una versione perfezionata del Chromebook. Tra le mani è una solida lastra in alluminio che ricorda il MacBook Air ma è realizzato in un metallo più liscio e un po' più morbido al tatto. Una volta aperto, il portatile rivela la sua struttura essenziale, elegante e funzionale. Dal bordo inferiore che circonda la tastiera ai tasti stessi e ai bordi del track pad, l'intera unità si ispira a un'estetica netta e squadrata, dettata dallo schermo 3:2. I bordi del dispositivo sono netti e gli danno un taglio estremamente moderno.



» Il Chromebook Pixel (2015) di Google ha potenza da vendere

USB-C come si deve

Le nuove porte USB-C sono una delle novità più importanti del **Chromebook Pixel**. La nuova porta USB 3.1 è una presenza notevole, alla luce del suo recente debutto sul nuovo MacBook. Tuttavia, diversamente dal più recente portatile di Apple, il Pixel è equipaggiato con due di questi nuovi ingressi in luogo di uno solo. Le due porte sono collocate ai due lati del laptop, il che vi consente di collegare l'alimentatore su un lato a vostra scelta. Sono ben pochi i Chromebook provvisti di uno schermo 'beyond HD', per non parlare dei 32 GB di spazio di archiviazione SSD o degli 8 GB di RAM di listino. Questo è di gran lunga uno dei più scattanti Chromebook sul mercato e non vi capiterà mai di rilevare rallentamenti, nemmeno dopo aver aperto venti o più schede in Google Chrome. Alimentato da un processore Core i5, il dispositivo dà il meglio di sé nella riproduzione in streaming di video a 1.080 p e perfino 4 K su YouTube. La maggior parte dei Chromebook se la cava bene con i video Full HD ma per lo più comincia a vacillare di fronte a risoluzioni superiori; il Pixel, invece, li riproduce in modo impeccabile. Se siete in grado di sopportare il suo formato 3:2, lo schermo del Chromebook Pixel è tra i migliori presenti sul mercato. I colori sono ricchi e l'abbondante contrasto consente di percepire pienamente i bianchi che sfumano in grigi delicati, fino al nero più profondo. Google ha perfino rimesso a punto la reattività e la precisione della funzione touchscreen. Detto ciò, siamo ricorsi di rado a questa funzione, se non per restringere e zoomare occasionalmente un'immagine su una pagina Web. L'autonomia della batteria del nuovo Chromebook Pixel dichiarata da Google è di 12 ore e questa stima non è molto lontana dalla realtà. Nelle nostre prove siamo riusciti a spremere 8 ore e 22 minuti con carichi di lavoro elevati. Quasi nove ore di autonomia sarebbero già notevolissime per un Chromebook ma il risultato è ancor più straordinario alla luce del processore Core i5 e dello schermo ad alta risoluzione che caratterizzano il Pixel. Google ha migliorato il Pixel quasi sotto ogni aspetto e soprattutto ha risolto i problemi di autonomia della batteria che caratterizzavano il suo primo laptop. Il Pixel è stato inoltre provvisto di uno schermo con funzione touchscreen più colorato e reattivo, nonché delle porte USB-C, il che ne fa un modello assolutamente da avere per chi non ha problemi di prezzo. LXP

Caratteristiche





USB-C

Nel complesso, le nuove porte USB-C rappresentano un vantaggio più che un limite per questo laptop.



Minimal

Speriamo che lo stile minimalista e la tastiera retro-illuminata vi piacciano quanto piacciono a noi.

LINUX PRO Giudizio

Chromebook Pixel (2015)

Produttore: Google
Web: www.google.com/chrome/devices
Prezzo: 1.128 €

Caratteristiche	8/10
Prestazioni	9/10
Facilità d'uso	10/10
Qualità/Prezzo	8/10

» *Il Pixel è il più veloce e il meglio progettato dei Chromebook... ma avete bisogno di questa potenza?*

Il voto di Linux Pro **9**

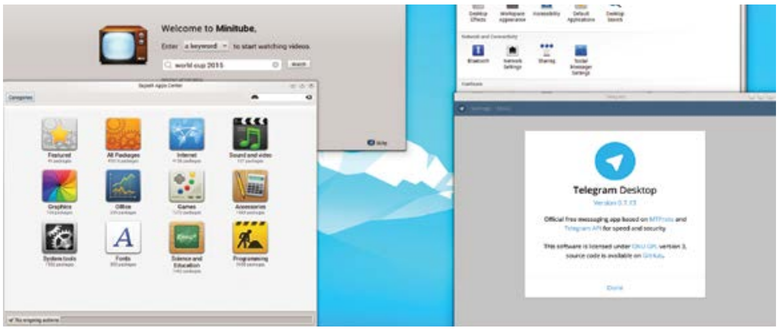
SuperX 3.0

Una delle tante distribuzioni KDE nate negli ultimi anni. Pur basandosi su presupposti piuttosto buoni, nel complesso non riesce a stupire più di tanto

In breve

» Una distro ben confezionata che dispone di una vasta scelta di applicazioni multimediali e Web. Basta su Ubuntu, questa distribuzione KDE pone particolare attenzione alle prestazioni e all'estetica. Per le alternative: OpenSUSE e Manjaro.

Negli ultimi anni, sono nate parecchie distribuzioni KDE pensate per i nuovi utenti. Alcune sono particolarmente orientate verso i principianti, altre rivolte a chi ha un bagaglio di conoscenze più consistente. **SuperX** può essere annoverata tra le prime. Questa distro è disponibile in versione Live per macchine a 32 e 64 bit. La ISO pesa meno di 2 GB. L'installazione fornisce alcuni termini di servizio (ToS), la cui accettazione è necessaria per continuare. Questi affermano esplicitamente che le licenze si applicano al software e a tutti i componenti prodotti da Libresoft, la società che sviluppa SuperX. Il tutto, però, non pregiudica i diritti concessi agli utenti dalle licenze del software incluso. Abbiamo però notato come i ToS non facciano menzione riguardo a software Open Source o a qualsiasi altro elemento che possa indicare le origini libere di questa distro. Pensiamo quindi di trovarci di fronte a un'omissione di una certa importanza. Per una distro che ha come obiettivo quello di fornire prestazioni superiori alle concorrenti, ci troviamo di fronte a un'installazione particolarmente lenta. Fortunatamente, il processo di partizionamento del disco è intuitivo e ben progettato. Potete scegliere se destinare l'intera unità a SuperX, oppure creare specifiche partizioni, con tanto di scelta se utilizzare LVM criptato o meno. Il processo d'installazione dura circa 30 minuti su una macchina dual core con 4 GB di RAM. Una volta installato, SuperX si è però dimostrato molto più veloce di molte altre distro KDE. In più, dispone di un'ottima collezione di applicazioni, utili a soddisfare le esigenze della maggior parte degli utenti. A questo proposito, i software Qt e KDE sono ben integrati, a differenza dei GTK che non stupiscono in quanto a prestazioni.



› **Una distro stabile e che fornisce una buona serie di programmi. Ciò nonostante non fa niente per distinguersi dai concorrenti**

Pronto all'uso

La distro presenta un sosia di Homerun chiamato **SuperX App Launcher**. Le applicazioni più usate, così come l'elenco dei file aperti di recente, sono subito disponibili. I programmi installati, invece, sono presenti nella scheda **Applicazioni**. SuperX può vantare un ottimo supporto multimediale che gli consente di far fronte a qualsiasi esigenza. L'ultima versione della distro utilizza il kernel Linux 3.13 e dispone del desktop KDE 4.13. Sulla carta, gli sviluppatori dichiarano che, grazie a una gestione della memoria migliorata, il sistema è molto più veloce rispetto ai principali concorrenti. Da parte nostra, non abbiamo notato alcuna differenza particolare. Va comunque detto che nell'uso di più applicazioni in contemporanea non si verificano particolari rallentamenti. Una nota positiva riguarda invece la stabilità. Durante l'utilizzo, non siamo incappati in alcun genere di crash. Per quanto riguarda la documentazione, dobbiamo segnalare una wiki piuttosto povera di contenuti, con informazioni non aggiornate e che avrebbero bisogno di una maggiore attenzione da parte degli sviluppatori e della comunità. Dall'altra parte, il forum ufficiale è abbastanza popolato e può fornire un valido supporto a chi ne ha bisogno.
In definitiva, anche se in questa distro non c'è niente che non vada, siamo rimasti piuttosto perplessi dalle note di rilascio. Infatti, siamo portati a diffidare per natura di quei sistemi che non dichiarano apertamente la propria natura Open Source. Infine, seppure la nuova versione arrivi quasi a due anni dall'inizio del suo sviluppo, sul sito non si trovano informazioni circa il modello di programmazione. LXP

Caratteristiche



Hub intrattenimento
La distro offre una vasta selezione di applicazioni multimediali che vi faranno divertire.



Derivata da Ubuntu
La distro è basata su Ubuntu, in modo da sfruttare dpkg e il sistema apt-get.

Linux Pro Giudizio

SuperX 3.0

Sviluppatore: Libresoft Tech. Pvt Ltd
Web: http://superxos.com
Licenza: GPL e altre

Caratteristiche	7/10
Prestazioni	8/10
Facilità d'uso	8/10
Documentazione	5/10

» SuperX non offre niente di speciale per i nuovi utenti e non aggiunge alcunché alle distro KDE.

Il voto di Linux Pro **6**

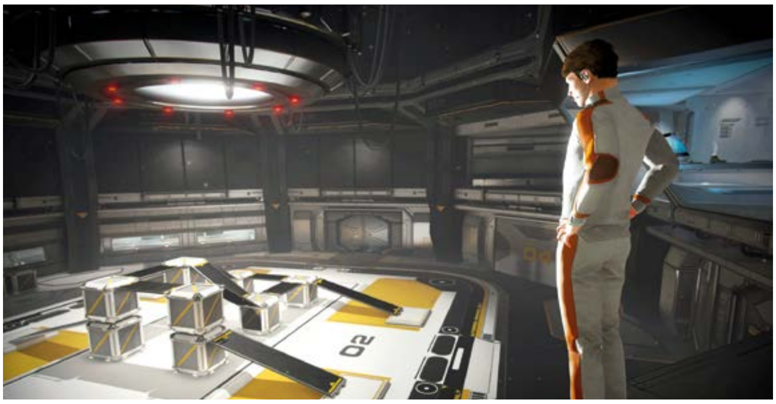
Unity 5

Dopo aver atteso per anni un motore di gioco di altissimo livello dobbiamo davvero dire che questa volta siamo di fronte e un vero campione...

In breve

» Independente dal sistema operativo
» Grafica AAA
» Editor 3D
» Editor 2D
» Strumento di animazione Mecanim
» Supporto C# e JavaScript
» Gratuito e senza pagamento di royalty
» Licenza a pagamento richiesta per i team
» Eccellente comunità e documentazione

Per gli sviluppatori di videogiochi indipendenti a corto di quattrini la situazione non è mai stata più propizia. Alla Game Developers Conference, Unreal Engine 4 è stato reso gratuito, Valve ha presentato il suo attesissimo motore di gioco Source 2 ed è stato pubblicato **Unity 5**. Unity è un motore per lo sviluppo di giochi che permette a sviluppatori e grafici di creare videogiochi utilizzando un'interfaccia grafica utente simile a quella che trovereste in un software di modellazione 3D, combinandola con C# o JavaScript per programmare il comportamento degli oggetti del gioco. Una volta creato, il gioco può essere esportato come eseguibile completo del motore necessario per gestire grafica, suoni, input dell'utente, animazioni, fisica e altro ancora. Unity 5 comprende diverse novità, per la prima volta a disposizione degli utenti della versione gratuita Personal Edition. In Unity 4, uno dei principali inconvenienti era che, in assenza di aggiornamenti a pagamento, l'opzione per le ombre soffuse era inaccessibile. Ciò costringeva lo sviluppatore indipendente ad aguzzare l'ingegno per ideare delle scappatoie, a tutto danno di un flusso di lavoro ordinato ed efficiente. Peggio ancora, per trarre veramente il massimo dalla grafica lo sviluppatore era costretto a imparare a programmare in *High Level Shader Language* (HLSL). In Unity 5 troviamo invece il sistema *Global Illumination* (GI) in tempo reale e un nuovo Standard Shader. Con lo **Standard Shader** potete impostare i materiali in modo tale che imitino praticamente qualunque materiale reale, il tutto senza creare una sola riga di codice. Queste due novità spalancano nuovi mercati e riducono gli ostacoli che finora impedivano agli sviluppatori indipendenti di integrare nelle loro creazioni un realismo di livello AAA.



› **Grazie alla riflessione indiretta della luce, vi basta impostare qualche sorgente luminosa e lasciare il resto a Unity**

Multi-supporto

Il motivo della grande diffusione di Unity è la sua capacità di adattare uno stesso codice a più sistemi operativi. Potete esportare il codice verso Linux, iOS, Android, Blackberry, Windows Phone 8, PC, OS X, Web, WebGL, Xbox 360, Xbox One, PS3, PS4, PS Vita e Samsung TV. Per quanto riguarda l'importazione dei contenuti, Unity 5 può importare i formati .FBX, .dae (Collada), .3DS, .dxf e .obj, nonché i formati proprietari. Mecanim, il sistema di animazione e rigging di Unity, permette di animare gli oggetti mediante clip di movimento importati da sorgenti esterne, creati con software terzi o acquistati da studi di motion capture. Peccato solo che in Unity non sia possibile creare animazioni personalizzate. I fenomeni fisici possono essere creati direttamente mediante le impostazioni per i collider, i corpi rigidi, le giunture e i vestiti; particolarmente azzeccata è la presenza di un motore Physics indipendente per i giochi 2D. Vi sono inoltre sistemi di particelle, un sistema per la creazione di alberi, strumenti per il terreno, un editor per gli sprite e strumenti per le interfacce (grafiche) utente. Insomma, non sembra mancare proprio nulla, anche considerato che è tutto quanto gratis. Nessun difetto, dunque? Beh, sebbene Unity sia in grado di esportare giochi che funzionano in Linux, l'SDK è compatibile solo con Windows e Mac. Qualcuno è riuscito a farlo funzionare anche in Wine ma il setup è un vero incubo. Inoltre, non sembra esserci gran che in termini di documentazione e supporto da parte della comunità. Il paragone con Epic, che ha iniziato a convertire UE4 per Linux e sembra offrire un notevole supporto ai propri utenti appare stridente. Per di più, GI e Standard Shader mancano di impostazioni predefinite di base. Particolarmente dolorosa per gli utenti poco avvezzi alla programmazione è l'assenza di strumenti di scripting visivi, come Blueprints di Unreal Engine o Flowgraph di CryEngine. La documentazione online, peraltro, è di buon livello. E se siete disposti a perdonare i suddetti limiti, c'è un'altra buona notizia: Unity ha accantonato il sistema di pagamento delle royalty... LXP

LINUX PRO Giudizio

Unity 5

Sviluppatore: Unity Technologies
Web: http://unity3d.com/5t
Prezzo: Gratuito; vers. Prof.75 $ al mese

Caratteristiche	9/10
Prestazioni	9/10
Facilità d'uso	8/10
Documentazione	9/10

» *Unity 5 potenzia il realismo con l'illuminazione, e l'applicazione delle luci alle texture in tempo reale.*

Il voto di Linux Pro **9**

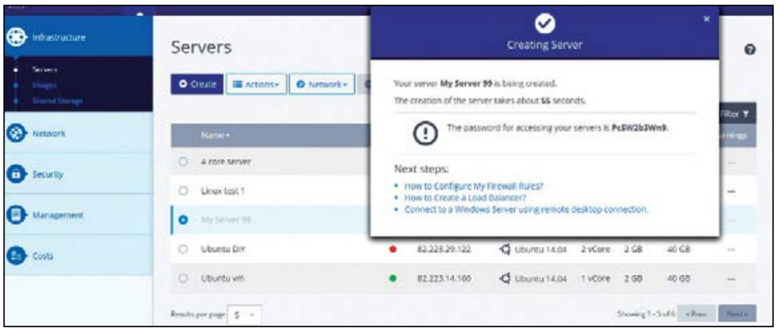
1&1 Cloud Server

Una piattaforma per il cloud scalabile e on-demand con tariffe al minuto che ha mandato in brodo di giuggiole la nostra redazione

In breve

» Un servizio di hosting su cloud robusto ma essenziale, con accesso a varie tipologie di macchine virtuali ospitate su VMware, bilanciamento dei carichi di lavoro e opzioni aggiuntive per l'archiviazione condivisa.

Cloud Server di 1&1 è progettato per gettare un ponte tra i servizi di hosting economici ma limitati e le piattaforme di fascia alta proposte da soggetti quali Amazon, Rackspace, ecc. E bisogna dire che lo fa con una certa classe, permettendovi di accedere a molteplici macchine virtuali ospitate su VMware per fare tutto ciò che volete, con tanto di applicazioni opzionali, bilanciamento dei carichi ed espansioni per la condivisione dello spazio di archiviazione; il tutto a un prezzo fisso ragionevole e senza alcuna clausola contrattuale relativa ad altri prodotti. Non è perfetto e ha qualche neo; ma nel complesso si tratta di un prodotto conveniente, scalabile e pratico che soddisfa abbondantemente le aspettative. Per il suo nuovo servizio Cloud Server, 1&1 ha creato vari centri dati dedicati negli Stati Uniti e, ciò che più conta per noi, anche uno in Europa. In tutti i casi, le componenti fisiche essenziali sono targate HP, con server che utilizzano hypervisor VMware supportati da fiammanti array di archiviazione SolidFire, il che permette alla casa di offrire 'di listino' ai suoi clienti veloci drive SSD invece che hard disk convenzionali. Il rovescio della medaglia è che questi dischi hanno una capacità massima di 500 GB ciascuno; d'altra parte, per ogni occorrenza del server cloud è possibile utilizzare fino a quattro dischi. Per le applicazioni particolarmente 'fameliche', è possibile utilizzare volumi condivisi ospitati su SAN fino a un massimo di 2 TB. Aggiungeteci il fatto che è sempre possibile collegarsi a un servizio di archiviazione su cloud dedicato qualora sia necessario ulteriore spazio.



› **Tramite il pannello di controllo potete sapere con precisione quanto vi costerà il vostro sistema di server**

Solide fondamenta

Per cominciare, vi collegate al vostro account 1&1 tramite il pannello di controllo Web dell'azienda, provvisto di un nuovo portale specificamente destinato a Cloud Server e chiamato, guarda caso, Cloud Panel. Appositi cursori vi permettono di controllare le tre opzioni principali e il prezzo da corrispondere viene aggiornato in tempo reale ogni volta che spostate i cursori. Sono disponibili macchine virtuali Windows e Linux; nel secondo caso, potrete scegliere tra Ubuntu, CentOS e Debian. Noi abbiamo creato una macchina virtuale a processore singolo con la configurazione minima di 1 GB di RAM e un drive SSD da 20 GB con CentOS 7 e la tariffa risultante è di circa 20 euro al mese IVA esclusa. Nel caso di Windows il prezzo sale, dal momento che la configurazione minima prevede un drive SSD da 40 GB più una licenza Windows Server. Inoltre, una caratteristica dei Cloud Server 1&1 è che spendete solo quando le macchine virtuali sono effettivamente in funzione: la 'bolletta' effettiva viene calcolata al minuto, il che permette di tagliare notevolmente i costi specie nel caso dei server usati per lo sviluppo. È inoltre possibile scegliere tra varie applicazioni, tra cui DQL Server e numerose piattaforme CMS; un'altra caratteristica notevole è la velocità con cui è possibile creare nuove occorrenze del server. Mentre con il precedente Dynamic Cloud Server di 1&1 questa procedura poteva richiedere ore, oggi basta premere il pulsante **Create** per rendere operative la maggior parte delle macchine virtuali in soli 55 secondi. Infine, abbiamo impiegato solo un paio di minuti per assegnare un po' di spazio di archiviazione condiviso SAN e per collegarlo ai nostri server di prova. Un'ultima opzione decisamente utile è la possibilità di controllare in qualunque momento quanto dovrete pagare per mezzo di una fattura interattiva. LXP

Caratteristiche



Setup rapido
Vi basta sottoscrivere un account Cloud Server a un costo basato sull'uso effettivo per mettere in funzione il vostro server in pochi minuti.



Costo al minuto
La configurazione flessibile, che applica i costi esclusivamente al minuto, permette di prevedere con precisione la spesa.

LINUX PRO Giudizio

1&1 Cloud Server

Fornitore: 1&1
Web: www.1and1.it
Prezzo: da 20 € al mese IVA esclusa

Caratteristiche	9/10
Prestazioni	9/10
Facilità d'uso	8/10
Qualità/prezzo	8/10

» *Potenza scalabile di elaborazione facile e veloce da installare: conveniente per le piccole aziende.*

Il voto di Linux Pro **9**

Fritz!WLAN Repeater

Abbiamo scoperto un modo facile per ampliare una rete wireless domestica

Il WLAN **Repeater 1750E** offre interessanti miglioramenti nel campo dei ripetitori LAN, il più importante dei quali è il supporto dual-band che consente di estendere una rete esistente da 2,4 GHz e 5 GHz dotandola di piene funzionalità 802.11ac. Finora noi usavamo un ripetitore Asus EA-N66 dual-band 3x3 802.11ac, che dava prestazioni eccellenti. Dopo averlo sostituito con Fritz!, abbiamo potuto immediatamente beneficiare di un allargamento della portata della rete, che ha raggiunto zone inaccessibili all'Asus. L'installazione è abbastanza semplice grazie alla tradizionale interfaccia Web di Fritz! e funziona a meraviglia; tenete conto però che il software client incluso supporta solo Windows e Mac. Selezionate le reti da 2,4 GHz e 5 GHz che desiderate estendere, aggiungeteci una password ed ecco fatto. Il sistema supporta anche semplici connessioni WPS. Una volta che lo avrete messo in funzione, la sua porta Gigabit Ethernet funzionerà anche come porta locale per qualsiasi dispositivo o router. Fritz! ha progettato il ripetitore con una spina a muro incorporata, il che può creare qualche inconveniente dato che l'unità misura 155 x 76,5 mm e ha una profondità di 62 mm. La parte anteriore presenta un indicatore relativo alla potenza del segnale, che tuttavia non può essere disattivato. Considerate le sue capacità, il dispositivo è conveniente in termini di prezzo rispetto alle soluzioni powerline e perfino ad alcuni ripetitori meno capaci. LXP



› **Aggiungete questa elegante unità alla vostra rete per ampliarne la portata**

LINUX PRO Giudizio

Fritz!WLAN Repeater 1750E

Sviluppatore: AVM
Web: http://bit.ly/WLANRepeater1750e
Prezzo: 83 €

Caratteristiche	8/10
Prestazioni	9/10
Facilità d'uso	8/10
Qualità/Prezzo	9/10

» *Offre progressi in termini di velocità della connessione e un reale aumento della portata della rete.*

Il voto di Linux Pro 8.5

Bioshock Infinite

Il gioco più bizzarro degli ultimi anni arriva su Linux...

Specifiche

» **Sistema operativo:** Ubuntu 14.10, Mint 17.1
» **CPU:** Intel Core 2 2,4 GHz+, AMD Athlon X2 2,7 GHz+
» **Memoria:** 4 GB
» **HDD:** 20 GB
» **Graphica:** Nvidia/AMD compatibile con Open GL 4.1, 1 GB di VRAM

BioShock Infinite è un gioco straordinariamente bello, la sua trama non è granché. È un gioco che merita assolutamente di essere provato da chiunque abbia la possibilità di farlo. Le scene sono splendide, a volte sbalorditive, tra le più sensazionali che ci sia mai capitato di vedere in un videogioco: e ora sono disponibili anche per gli utenti di Linux. Ambientato nell'universo di Bioshock, il gioco abbandona le profondità marine per dare la scalata ai cieli, proiettandovi tra le nuvole in un mondo follemente fanatico e ultra-razzista in cui dovrete salvare la situazione nonché la classica damigella in pericolo. Non è un gioco recente, dato che risale al marzo 2013; a due anni di distanza, eccolo uscire per Linux con una versione aggiornata del motore eON, che tanti problemi ha creato alla versione per Linux di Witcher 2. La buona notizia è che di questi problemi non c'è alcuna traccia in BioShock Infinite, che anzi ci ha offerto una delle esperienze di gioco più fluide da noi sperimentate negli ultimi tempi. Il che è notevole, considerate le strabilianti immagini che il gioco regala perfino con una scheda video limitata come la Radeon HD 7750 da noi utilizzata per la nostra prova. Non appena lo troverete in vendita online, non perdete l'occasione di accaparrarvi uno dei migliori sparatutto in prima persona del decennio. LXP



› ***Bioshock Infinite*** **è il titolo AAA per Linux da avere per forza**

LINUX PRO Giudizio

Bioshock Infinite

Sviluppatore: Virtual Programming
Web: http://store.steampowered.com
Prezzo: 28 €

Giocabilità	9/10
Grafica	10/10
Longevità	8/10
Qualità/Prezzo	9/10

» *Una trama intricata e spettacolare, dà vita a un gioco divertente, con ottimi contenuti aggiuntivi.*

Il voto di Linux Pro 9

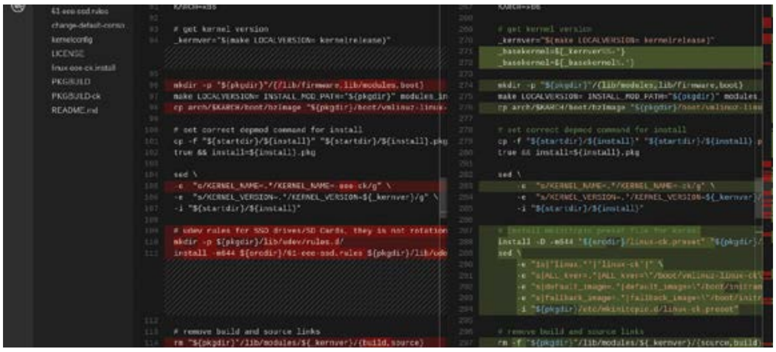
Visual Studio Code

Microsoft fa qualcosa per Linux?! Vogliamo vederci chiaro...

In breve

» Un editor di codice per tutti i sistemi operativi che combina tecnologie Open Source (Chromium, io.js, Typescript ed Electron) e IntelliSense, il sistema di completamento automatico del codice di Microsoft. Leggermente più avanzato di Geany o Juffed, è però ben lontano dal costituire un ambiente di sviluppo in grandezza naturale quali Eclipse o Visual Studio. Vedi anche: Light Table, vi.

Sotto Satya Nadella, Microsoft sembra aver assunto un atteggiamento molto più aperto nei confronti di Linux. L'ultimo contributo dell'azienda è rappresentato da un editor di codice compatibile con tutti i sistemi operativi (Linux compreso) chiamato **Visual Studio Code**, presentato da Microsoft durante la conferenza Build dell'aprile 2015. Con il lancio di questo prodotto Redmond afferma di voler andare incontro agli sviluppatori "ovunque si trovino". Chi si aspetta una riedizione in grandezza naturale del colosso IDE che risponde al nome di Visual Studio rimarrà deluso. Stiamo parlando infatti di un semplice editor 'leggero' e, francamente, nemmeno eccezionale. All'avvio iniziale vi trovate di fronte un'area di lavoro grigio scuro, sotto la quale si trova una barra di stato che mostra una sfumatura di viola particolarmente insolita. Si tratta del cosiddetto Dark Theme; se volete, potete passare al Light Theme semplicemente cliccando su un pulsante. Selezionando le impostazioni utente potete intervenire su qualche altra opzione, ma la procedura è quanto di più lontano dallo stile Microsoft: si apre infatti un file settings.json vuoto accanto a quello che riporta le impostazioni predefinite, che potete così scegliere di ignorare. La vera sorpresa, tuttavia, è che l'intera interfaccia è renderizzata mediante un browser Chromium modificato. Di fatto, tutto quanto è costruito intorno a un framework Open Source sviluppato da GitHub di nome Electron, che utilizza io.js e altre tecnologie Web per creare applicazioni autenticamente indipendenti dal sistema operativo. Come i più scettici avranno già intuito, tuttavia, Visual Studio Code è un software proprietario. Si dà il caso che anche Adobe abbia pubblicato un editor, Brackets, più orientato allo sviluppo Web ma si tratta di un prodotto Open Source.

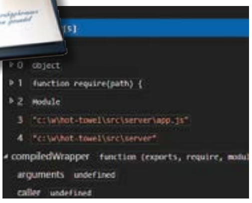


› **Confrontare i file fianco a fianco è più pratico che giocare a 'trova le differenze'**

'Leggere' anche le funzioni

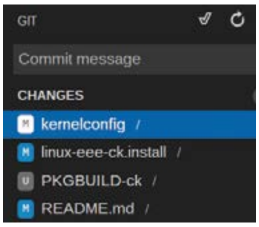
Il sistema di evidenziazione della sintassi di Visual Studio Code supporta una trentina di linguaggi, cioè decisamente meno rispetto a Kate, Gedit, JuffEd, Geany o qualunque altro editor di codice che valga la pena di citare. E solo per alcuni di questi linguaggi sono disponibili le funzioni più avanzate. D'altro canto, stiamo parlando di una versione di anteprima del software, per cui sotto questo aspetto la situazione è destinata a migliorare. La funzione di confronto del codice (foto in alto) e il supporto Git sono senz'altro utili e non le troverete in molti editor di codice leggeri. Tuttavia, molti sviluppatori utenti di Linux preferiranno servirsi di diff e di git dalla linea di comando. E anzi, potendo scegliere, molti opteranno piuttosto per vi o Emacs. Tra l'altro, qui il concetto di 'leggerezza' si applica solo alle funzioni, non alle dimensioni del programma. Una volta decompressa, l'installazione 'pesa' ben 220 MB, sebbene gran parte di questa mole sia dovuta a una grossa libreria di Chromium. Il supporto per Visual Studio Code è stato introdotto nella versione 0.7 di Ubuntu Make, perciò chiunque utilizzi l'ambiente di Canonical potrà servirsi di questo programma. Nel momento in cui scriviamo, il sistema di debugging supporta solo le applicazioni Node.js e ASP, offrendo però tutte le funzioni di impostazione dei breakpoint, analisi delle variabili e stepping del caso. Il che non è sufficiente, tuttavia, a indurci a consigliarvelo rispetto agli innumerevoli editor di testo Open Source con funzioni di elaborazione del codice. Se proprio siete alla ricerca di novità, provate **Light Table**. LXP

Caratteristiche



Debugger

La funzione interna di debugging può alleggerire molto il carico di lavoro dello sviluppatore... ma supporta solo ASP e Node.js.



Supporto Git

Aprite un repository Git e Visual Studio Code vi indicherà tutte le nuove modifiche.

LINUX PRO Giudizio

Visual Studio Code

Sviluppatore: Microsoft
Web: https://code.visualstudio.com
Licenza: Freeware (prod. proprietario)

Caratteristiche	6/10
Prestazioni	6/10
Facilità d'uso	6/10
Documentazione	7/10

» *Esistono già soluzioni gratuite e Open Source molto superiori, quali che siano le vostre esigenze.*

Il voto di Linux Pro **6**

Il confronto

Ogni mese mettiamo a confronto prodotti e programmi per farvi scegliere al meglio!

Instant messaging

Passate dai programmi di IM proprietari a quelli free. Linux Pro ne ha selezionati cinque: ecco qual è il migliore...



Modalità del test

Un buon programma di Instant Messaging è discreto e non legherà né voi né i vostri contatti a qualsiasi software, dispositivo o servizio. Con queste caratteristiche ben precise in mente, ci siamo concentrati sull'IM testuale. Per utilizzare qualcosa che non viene bloccato, dovrete scegliere di usare un account XMPP e il relativo software. L'*eXtensible Messaging and Presence Protocol* (**http://xmpp.org**) utilizza XML per fornire le funzioni di messaggistica in tempo reale e gestione dei contatti. I messaggi XMPP devono passare tramite un server che usa lo stesso protocollo, garantendo così l'intercomunicazione. Abbiamo scelto di usare cinque client XMPP che potete trovare nella maggior parte delle distribuzioni. Ci siamo quindi concentrati sulla messaggistica testuale, anziché sulle funzioni audio e video.

Al giorno d'oggi, molte persone non vedono oltre programmi come WhatsApp, Facebook Messenger, Google Talk e Skype. Noi, però, siamo di un'altra razza e ci piace pensare che se veramente amate l'Instant Messaging di tipo testuale, allora dovreste imparare a usare almeno uno dei cinque programmi che presentiamo in questo confronto. Prima di tutto, non forzano nessuno a utilizzare la stessa interfaccia o lo stesso provider. Rispetto ai tradizionali servizi on-line di cui abbiamo parlato all'inizio, questi software fanno parte della schiera dei programmi in piena regola e quindi sono molto più flessibili e versatili da configurare. Inoltre, le comunicazioni che avvengono tramite i messaggi possono essere utilizzate senza problemi anche su reti private. Avrete poi la possibilità di importare ed esportare il vostro elenco dei contatti, le trascrizioni complete delle conversazioni, nonché tutta un'altra serie di funzioni molto utili. Infine, per ogni programma scelto, avrete a disposizione un gran numero di tutorial disponibile in Rete. Senza poi dimenticare il supporto tecnico online e una vasta serie di articoli per utilizzare al meglio ogni strumento del software.

> **"Le comunicazioni IM possono essere usate senza problemi anche su reti private"**

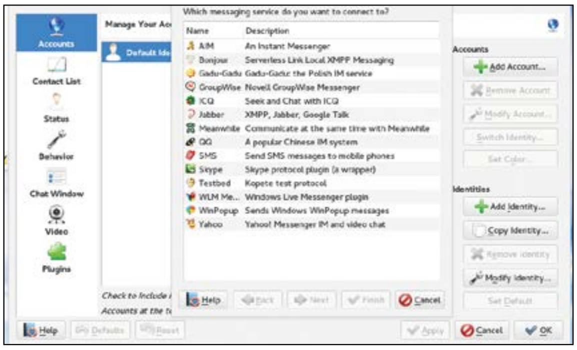
La nostra selezione

- Empathy
- Jitsi
- Kopete
- Pidgin
- Psi

La gestione dell'account

Potete configurare senza pensieri le vostre rubriche di contatti?

Tutti i programmi presi in considerazione sono in grado di gestire più account XMPP che già esistono su qualsiasi server che utilizza questo protocollo. Ciò include Facebook e Google Talk, naturalmente fino a quando si seguono le istruzioni ufficiali di ciascuna azienda per client di messaggistica generici. La creazione di nuovi account XMPP attraverso ogni client viene pienamente supportato, ma in pratica funziona solo se il server lo consente. Se non avete bisogno di più account, basta configurare il programma in questione con pochi click del mouse. **Psi**, in questo caso, cercherà anche di rimuovere il rispettivo profilo dal server corrispondente. Tutti i client, a eccezione di Psi, sono in grado di gestire account non XMPP. Basta solo qualche impostazione e l'inserimento dei dati da parte nostra. L'elenco di tutti i servizi supportati da ogni programma è assurdamente lungo. Potete utilizzare AIM, Gadu Gadu, ICQ, MySpace, MXit, Zephir e perfino il buon vecchio Internet Relay Chat (IRC). **Jitsi** fornisce un insieme di servizi supportati tra i più diversificati. Per esempio, questo client riconosce sia SIP sia iptel.org. Tutte le operazioni di gestione degli account (creazione, configurazione e rimozione) che abbiamo creato con i cinque programmi sono andate a buon fine. C'è solo un'eccezione: non siamo riusciti a rimuovere un account da Psi per un non meglio specificato "evento in corso". In termini di reti, **Pidgin** e il suo esercito di plug-in vince a mani basse. Tuttavia, a nostro avviso, **Kopete** offre il miglior compromesso tra reti supportate e la gestione efficiente dei rispettivi account. Quando si utilizza Kopete, per esempio, si devono prima di tutto creare identità specifiche per il tipo contatto (famiglia, amici, lavoro, ecc.), quindi ogni specifica identità deve essere associata con un numero di account in una delle reti disponibili. Per impostazione predefinita, come dati personali, Kopete supporta solo email, telefono privato e cellulare. Jitsi, invece, dispone di parecchi campi in più.



› **La gestione degli account di *Kopete* richiede prima la creazione di un'identità e poi l'associazione agli account**

Verdetto

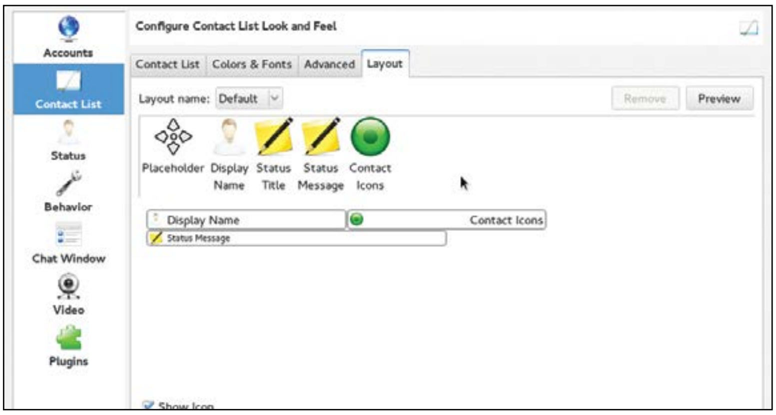
Empathy ★★★☆☆
Jitsi ★★★★☆
Kopete ★★★★★
Pidgin ★★★★☆
Psi ★★★☆☆

» *La gestione degli account e delle identità fa di Kopete il vincitore.*

Gestione dei contatti

Una lista ben organizzata è un elemento irrinunciabile

Come si fa a gestire la rubrica dei vostri contatti? L'aggiunta di un profilo non è un problema, ma ci sono comunque parecchie differenze tra i vari client che permettono di visualizzare i contatti in modo diverso. Ciò nonostante, ogni programma permette di gestire gli alias e il raggruppamento. Kopete condivide i suoi contatti con Kmail, Kontact e altre applicazioni tramite lo stesso formato. Inoltre, può essere definito il migliore per quanto riguarda la visualizzazione. L'interfaccia che elenca i contatti ha quattro schede, ognuna completa di una serie di opzioni. Per aggiungere funzioni accessorie, potete poi usare il plug-in **Contact Notes** che integra ulteriori informazioni personali a ciascun contatto. Kopete e Pidgin possono inserire in un meta-profilo i gruppi di account che, pur corrispondendo alla stessa persona, utilizzano reti diverse. Da considerare poi che Pidgin e Jitsi possono trattare i feed RSS come contatti di IM. In altre parole, è possibile vedere sotto forma di messaggio istantaneo qualsiasi news proveniente da un flusso RSS. Non è necessariamente una caratteristica indispensabile, ma può essere considerata molto utile.



› **Con *Kopete* è possibile riorganizzare la posizione di ogni voce che può essere inclusa nella lista dei contatti. Basta una semplice azione di drag-and-drop**

Verdetto

Empathy ★★★☆☆
Jitsi ★★★★☆
Kopete ★★★★☆
Pidgin ★★★★☆
Psi ★★★☆☆

» *Kopete vince, anche se bisogna stare attenti alla configurazione.*

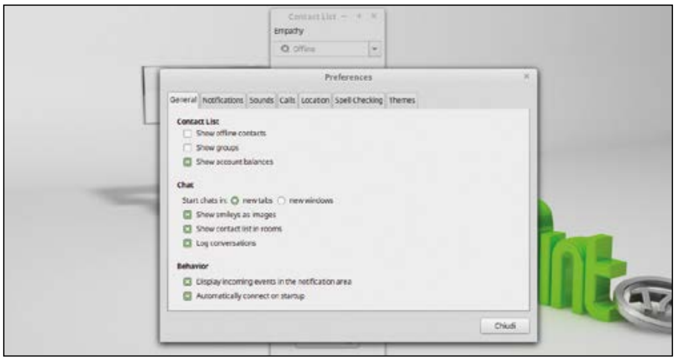
Interfaccia utente

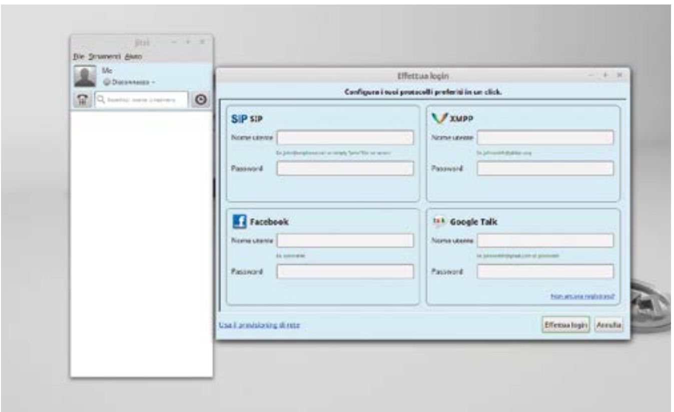
Quanto, ma soprattutto perché, si desidera usare l'IM?

Ci sono persone che, trattando di programmi di messaggistica istantanea, non si preoccupano dell'interfaccia. In generale, però, non dovrebbe essere questo l'approccio con cui ci si avvicina a queste applicazioni. Naturalmente, avere una buona GUI non deve essere l'unico pensiero, ma è fondamentale tenere questo aspetto in buona considerazione. Soprattutto chi usa un software di IM per lavoro o comunque per lunghi periodi vuole avere un ambiente curato e piacevole da guardare. Inoltre, oltre all'estetica, è importante la praticità che vi consentirà per esempio di gestire più conversazioni insieme.

Empathy ★★★☆☆

Empathy è il client di default per Gnome e non fa niente per nasconderlo. In effetti, è così integrato che al primo avvio vi trovate a fissare una finestra vuota al centro del desktop. Non ci sono elementi da gestire e non si può fare a meno di chiedersi se non si sia sbagliato qualcosa nel processo di avvio o tutto stia funzionando a dovere. Basta però accorgersi della comparsa di un piccolo widget nella barra delle applicazioni. Ci si rende quindi conto che Empathy viene gestito da lì. L'interfaccia è pulita e ha tutte le funzioni principali a portata di mano. In più dispone anche di un bel look che potete cambiare con più temi (**http://bit.ly/EmpathyThemes**). È un client essenziale per chi non vuole perder tempo.





Jitsi ★★★★☆

Questa applicazione in Java funziona bene anche su Windows e Mac OS. Ha un aspetto un po' "aziendale" e lo consideriamo un programma troppo basato sugli aspetti audio e video. Questo rende Jitsi meno *user friendly* rispetto alla media. La verità è che tutte le sue funzioni aggiuntive sono molto più complesse e meno standardizzate rispetto agli altri. Infatti, il software contiene un'enorme varietà di opzioni aggiuntive, ognuna delle quali vi richiederà di conoscere almeno un termine legato alla telefonia. Ciò nonostante, ci sono anche parecchie caratteristiche molto interessanti, come la buona interfaccia per la ricerca delle chiamate e il pannello di configurazione per le scorciatoie da tastiera. Se vi prendete un po' di tempo per fare la sua conoscenza, Jitsi può rivelarsi un ottimo programma di IM.

Integrazione desktop

Come si comportano sui tradizionali desktop?

Tutti i software di questo confronto si integrano piuttosto bene con l'ambiente desktop. Da questo punto di vista, Empathy e Pidgin si trovano meglio con Gnome, mentre Kopete, come avrete intuito fin dall'inizio, si adatta perfettamente a qualsiasi ambiente KDE. Psi è così compatto che il suo aspetto è davvero poco importante. Jitsi, invece, è il programma che più di ogni altro sembra alieno e fuori dal contesto dei desktop tradizionali. Apparenze a parte, la cosa più importante da valutare sono le notifiche, ossia le piccole finestre di dialogo che compaiono brevemente per visualizzare i nuovi messaggi. Quanto bene riescono a lavorare su Linux dipende essenzialmente dal windows manager e dal desktop utilizzato. A parte questo, tutti i programmi, fatta eccezione per Jitsi, hanno il loro sistema di notifica. Ognuno ha lavorato piuttosto bene sia su Gnome sia su KDE, seppure Pidgin e Kopete si distinguano dalla massa per alcuni motivi. In primo luogo, i pop-up **libnotify** di Pidgin consentono di impostare la combinazione di nuovi messaggi e conversazioni, nonché avvisi provenienti da contatti offline. Inoltre, utilizzando il plug-in **Higlight**, possiamo impostare i programmi per essere disturbati solo quando in un messaggio vengono rilevate espressioni particolari. Kopete, infine, può anche registrare alcuni messaggi ed eseguire uno script personalizzato.

Verdetto

Empathy ★★★★★

Jitsi ★★★☆☆

Kopete ★★★★☆

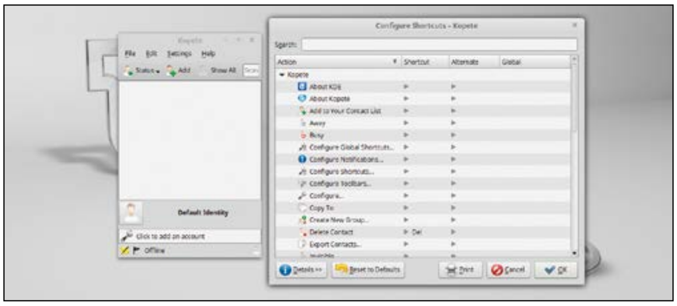
Pidgin ★★★★☆

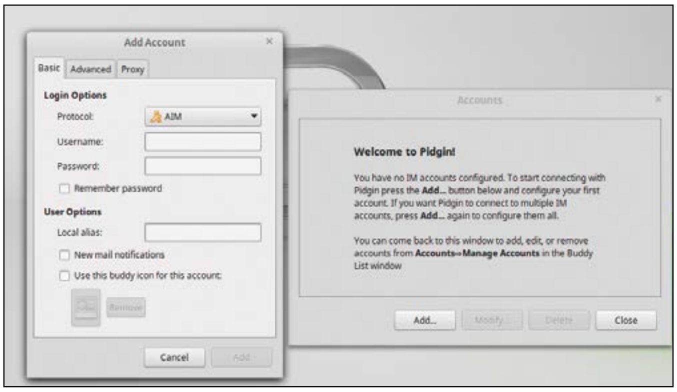
Psi ★★★☆☆

»Kopete e Pidgin offrono entrambi un'ottima serie di funzioni ben integrate.

Kopete ★★★★☆

Kopete non può essere definita un'applicazione elegante, ma mira soprattutto a essere funzionale e flessibile. È possibile personalizzare quasi qualsiasi aspetto, a patto però di avere un po' di pazienza. Lo stile della finestra di chat, per esempio, può essere modificato con XSL e fogli di stile CSS. In più, niente vieta di personalizzare le emoticon con un proprio stile. Anche il pannello per la configurazione delle notifiche è eccezionale. I plug-in ci sono e forniscono funzioni assai utili: dalla tradizione automatica al controllo ortografico.



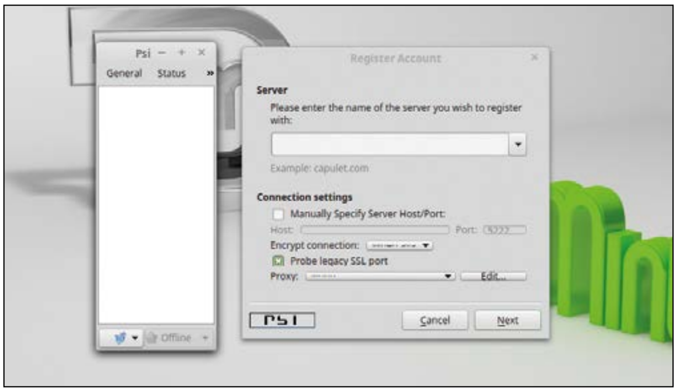


Pidgin ★★★★★

L'interfaccia utente predefinita di Pidgin è ordinata e facile da consultare. Se non trovate una funzione che faccia al caso vostro, è molto probabile che ci sia un plug-in che permette di integrarla. Attenzione però, alcune voci elencate nel sito sono completamente scomparse dalla versione per Linux, rimanendo invece disponibili per Windows. La release per il Pinguino, tuttavia, consente di utilizzare uno strumento a riga di comando. Gli sviluppatori descrivono poi un'altra opzione utile all'uso di macro e azioni registrate. Queste possono essere applicate velocemente e quando ne abbiamo bisogno. Per esempio, è possibile impostare un messaggio predefinito da inviare a un contatto ogni volta che il suo status cambia.

Psi ★★★☆☆

Psi è un client interessante da usare, ma anche il più deludente del gruppo. Tra gli aspetti positivi c'è l'ottima localizzazione che permette di usarlo in ben 20 lingue. In più, è multipiattaforma e può essere sfruttato su vari sistemi operativi. A livello di flessibilità è molto simile a Kopete. Il layout della finestra di chat rappresenta poi un buon compromesso tra compattezza e funzionalità. Inoltre, è possibile gestire l'interfaccia a piacimento, variandone i colori e le emoticon. Purtroppo, però, ci sono alcuni aspetti che rendono l'uso di Psi poco pratico. Uno di questi è l'impossibilità di comprimere l'elenco dei contatti di Facebook. Prima di aggiungere altri insiemi di account, è necessario scorrere tutta la lista di amici.



Chiamate voce e video

Potete parlare faccia a faccia quando il testo non basta?

Anche se ci siamo concentrati sullo scambio in tempo reale di messaggi, ciò non toglie che esistano anche le chiamate audio e video. Avere a che fare con software che ne supportano la tecnologia, rende la vostra scelta più semplice, soprattutto in base all'uso principale che ne volete fare. Pidgin e Empathy hanno, almeno sulla carta, il supporto per tali funzioni. Tuttavia, il loro uso dipende troppo da svariate combinazioni come le versioni del software e la configurazione di tutti gli account IM coinvolti. Se si desidera utilizzare questi due client per l'audio e il video, dovrete perdere un po' di tempo nella ricerca di informazioni online. Se quindi avete intenzione di effettuare chiamate extra-testuali con Software Libero, per questo confronto vi consigliamo di dare un'occhiata a Jitsi. Si tratta senza dubbio del programma più indicato per questo genere di attività. Le chiamate voce e la crittografia video utilizzano una tecnologia chiamata **ZRTP** (vedi **https://jitsi.org/Documentation/ZrtpFAQ**), sviluppata dallo stesso autore di PGP. Potreste anche essere in grado di usare Jitsi per parlare con utenti che sfruttano altre reti come SIP o Google Chat, ma non è detto che la comunicazione riesca. In altre parole, per esserne sicuri dovrete prima provare.

Verdetto

Empathy
★★☆☆☆

Jitsi
★★★★☆

Kopete
★★★☆☆

Pidgin
★★★☆☆

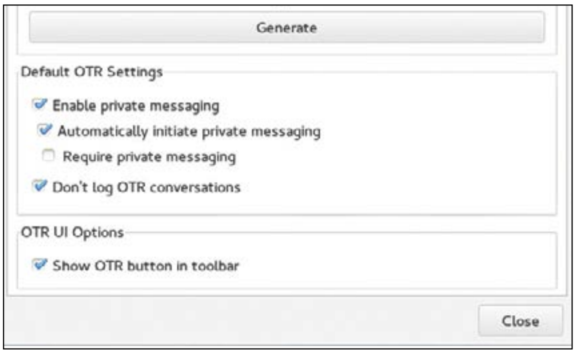
Psi
★★☆☆☆

» Per chi ha bisogno di fare chiamate voce e video, Jitsi è la scelta migliore.

Sicurezza

Fate attenzione, non si sa mai chi può essere in ascolto!

Come già accennato, un motivo per usare la messaggistica istantanea testuale consiste nello sfruttare una maggiore privacy rispetto a Skype e WhatsApp. Per avere questa sicurezza, è però necessario soddisfare almeno due requisiti. Il primo e il più difficile è convincere gli altri che si vuole parlare in privato e quindi occorre installare il software giusto, nonché procedere all'installazione e alla configurazione. In seconda battuta, bisogna ricordarsi di non lasciare tracce della conversazione avuta sul computer utilizzato. Empathy, sotto questo profilo, è la scelta peggiore poiché supporta solo la crittografia SSL. Infatti, almeno in questo ultimo periodo, la scelta dei client IM verte più su OTR (*Off The Record messaging*) che funziona nello stesso modo generico di PGP e GPG. Le procedure per la configurazione della crittografia sono comunque ben documentate e basta solo un po' di pazienza per metterle in pratica senza alcun problema. Le password degli account sono un ulteriore elemento da valutare. Su questo fronte, Jitsi si distingue come il client che memorizza una password per ciascun account, ma anche una chiave master per usare tutte le altre credenziali di accesso.



› **L'interfaccia di configurazione di *Pidgin* permette di mettere mano alle funzioni OTR**

Verdetto

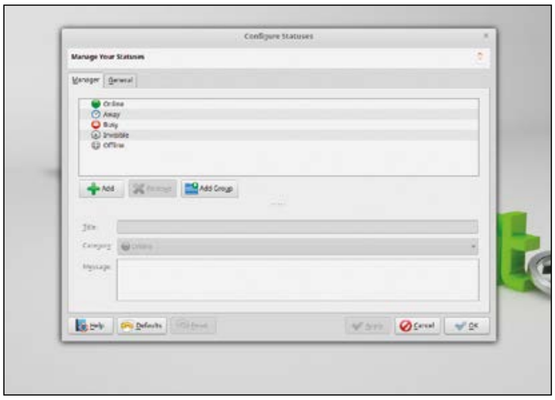
Empathy
★☆☆☆☆
Jitsi
★★★★☆
Kopete
★★★★☆
Pidgin
★★★★☆
Psi
★★★☆☆

» *OTR è lo standard per la crittografia nei client IM Open Source. Pidgin è quello che permette di utilizzarlo al meglio.*

Gestione delle informazioni

Permettono di lavorare in modo efficiente?

Uno degli aspetti più importanti di un client IM è il controllo che si ha nella gestione degli stati, vale a dire quando vogliamo far sapere o meno se siamo in linea o se occupati e non possiamo rispondere ai messaggi. Tutti i programmi presi in considerazione permettono di impostare vari stati, nonché di crearne nuovi e personalizzati per adattarsi alle varie esigenze. Pidgin, inoltre, offre anche una buona selezione di "stati d'animo" e quindi consente di far sapere ai vostri contatti come vi sentite in un dato momento. Se impostate su "Nervoso", forse è meglio se vi stanno alla larga. A parte questo, è però importante precisare un aspetto fondamentale. La gestione delle informazioni, in questo confronto, può essere considerata un termine sotto il quale devono poi essere inseriti altri aspetti, come la gestione generale del flusso di dati, nonché quando è possibile inviare e ricevere durante una sessione di chat. In questo senso, pertanto, possiamo darvi solo un'idea generale di come stanno le cose, dal momento che la risposta varia in base alla combinazione di client, ambiente desktop e account XMPP utilizzato per la conversazione. Le emoticon, per esempio, non sono un aspetto complesso, ma possono essere visualizzate in modo diverso in base al client che si utilizza. Il trasferimento di file e le notifiche, comunque, funzionano bene su tutti i programmi a prescindere dalla configurazione e dal sistema utilizzato. Lo stesso vale per la creazione e l'utilizzo di chat di gruppo o chat room, a patto però che tutti i contatti coinvolti utilizzino account XMPP generici e sfruttino la stessa rete. Tutti i client permettono poi di portare avanti discussioni simultanee nella propria scheda, nonché avere la possibilità di personalizzare la finestra della chat in diversi modi. Psi, per esempio, dispone di un pannello di configurazione per impostare diversi colori, utile a individuare le parti rilevanti di una conversazione. I temi di Empathy, invece, consentono di modificare completamente l'aspetto del programma. Nella nostra esperienza, infine, i migliori client per la gestione delle informazioni sono Pidgin e Kopete. Con entrambi è possibile controllare quali contatti sono configurati per una conversazione criptata, nonché personalizzare le varie chat room.



› **La maggior parte dei client di messaggistica istantanea permette di impostare status personalizzati da far visualizzare ai vostri contatti**

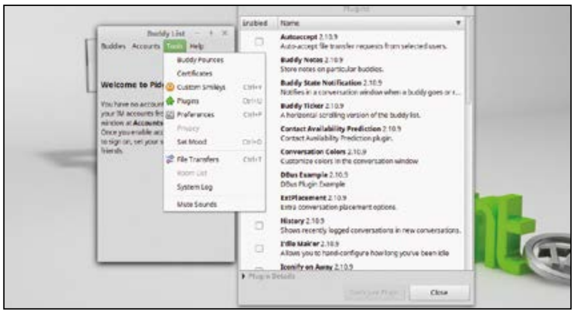
Verdetto

Empathy
★★★☆☆
Jitsi
★★★★☆
Kopete
★★★★★
Pidgin
★★★★★
Psi
★★★☆☆

» *Prendetevi il tempo per configurarlo e Pidgin renderà le conversazioni più efficienti.*

Instant Messaging
Il Verdetto

Gli account XMPP sono gratuiti e non c'è bisogno di utilizzare alcun indirizzo di posta elettronica per registrarli. I client desktop XMPP, inoltre, sono disponibili per tutti i sistemi operativi. Ci sono anche applicazioni dedicate ai dispositivi mobile. Il motivo per cui questi programmi non vengono utilizzati e non sono diventati fenomeni di massa è dovuto soprattutto alla moda. Client come Facebook e WhatsApp sono oramai diventati fenomeni enormi che difficilmente verranno abbandonati per altre soluzioni. Il loro punto forte, infatti, è togliere l'imbarazzo all'utente di inserire le proprie credenziali. Un problema più grave di cui soffrono le nostre applicazioni consiste nell'interoperabilità. L'IM, infatti, non è un metodo universale per scambiarsi messaggi digitali. Immaginate cosa succederebbe se la formattazione delle email dipendesse esclusivamente dal tipo di client utilizzato per la gestione della posta elettronica. Sarebbe un vero caos. In altre parole, nell'IM mancano gli standard che permettono a questo sistema di diventare univoco e indipendente dalla piattaforma sfruttata. Dovendo stilare una classifica, abbiamo deciso di mettere Pidgin al primo posto. Kopete, invece, arriva secondo, ma solo perché può essere più difficile da approcciare per i principianti. Piding, inoltre, dispone di una grande quantità di plug-in che permettono di implementare parecchie funzioni molto utili. Empathy e Psi sono buoni software, ma il secondo non è abbastanza stabile come avremmo voluto. Inoltre, la mancanza di supporto per OTR non può passare inosservata. Se poi avete bisogno di un'alternativa a Skype per la comunicazione voce o video, vi consigliamo Jitsi.



› ***Pidgin* è quanto di meglio si possa chiedere: supporta OTR, ha tonnellate di plug-in e parecchie emoticon**

> **"Pidgin si distingue per flessibilità e praticità d'uso. In più è molto configurabile"**

1° Pidgin ★★★★☆
Web: https://pidgin.im **Licenze:** GPL 2+, GPL 2, MIT **Versione:** 2.10.11
» Facile da usare, ma anche molto potente.

2° Kopete ★★★★☆
Web: http://kopete.kde.org **Licenze:** GPL 2+ e GFDL **Versione:** 0.12.2
» Potente e solido, ma molte opzioni sono superflue.

3° Jitsi ★★★★☆
Web: www.jitsi.org **Licenza:** LGPL **Versione:** 2.9
» Ottime caratteristiche, anche se indirizzate soprattutto all'audio e video.

4° Psi ★★★☆☆
Web: http://psi-im.org **Licenza:** GPL 2+ **Versione:** 0.15
» Non completo come i primi in classifica, ma nel complesso è buono.

5° Empathy ★★★☆☆
Web: http://live.gnome.org/empathy **Licenza:** GPL 2+ **Versione:** 3.12.3
» Buon programma che però non supporta OTR.

A voi la parola...
Non siete d'accordo con le nostre scelte? Avreste usato altri software? Inviate le vostre opinioni su questo confronto a: **recensioni@linuxpro.it**

Considerate anche...

Nell'enorme panorama dell'Open Source, ci sono davvero tanti software per la messaggistica istantanea dedicati a Linux. Questo è comunque un bene, perché vi consente di scegliere con maggiore libertà in base alle vostre esigenze. La lista semi ufficiale di tutti i software IM la potete trovare a questo indirizzo: **https://xmpp.org/xmpp-software/clients**. Controllando questa fonte, date particolare rilevanza alle applicazioni per Android e iOS che vi consentono una maggiore flessibilità d'uso. Convincendo i vostri amici, parenti e colleghi a utilizzarli, potreste liberarvi dalle grinfie del software proprietario. Naturalmente, se volete garantirvi la massima privacy, date la preferenza ai programmi che supportano OTR con chiavi temporanee. L'altra cosa importante da considerare è l'uso di un server XMPP. Se avete bisogno di un client di messaggistica istantanea molto sicuro, basta usare uno dei server elencati a questo indirizzo: **https://xmpp.org/xmpp-software/servers**. LXP

Da non perdere

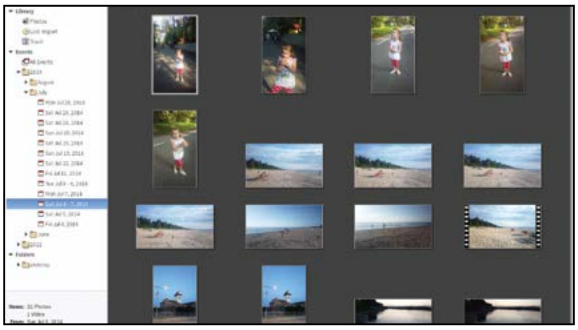
Shotwell » Linphone » Fiziko » Audacity » Brackets » Netsurf » Geary » Pasang Emas » FreeCol » VirtualBox » FreeMind

Gestore fotografico

Shotwell

Versione: 0.22 **Web:** http://bit.ly/GnomeShotwell

Shotwell ha esordito nel 2009, quando ha sostituito F-Spot come applicazione predefinita per la gestione delle foto in Gnome. Si tratta di un'applicazione snella e ordinata, realizzata in GTK in modo appropriato e semplificato. Il suo scopo è fornire tutti i servizi più essenziali per l'organizzazione delle vostre immagini. Shotwell può importare foto e video direttamente dalla fotocamera tramite la modalità di archiviazione di massa USB o MTP. Una volta importati, i file vengono archiviati nella directory scelta, quindi Shotwell provvede a raggruppare tutti i file suddividendoli per data. La visualizzazione è simile a quella di un qualsiasi file manager ed è possibile sfruttare anteprime di dimensioni diverse per capire subito quale foto volete aprire. La barra di sinistra serve per la navigazione tra gli album, mentre l'area principale mostra il contenuto delle cartelle selezionate. Shotwell viene quindi utilizzato come strumento per la navigazione tra i file digitali, grazie soprattutto alla sua flessibilità e alla capacità di gestire sia documenti video

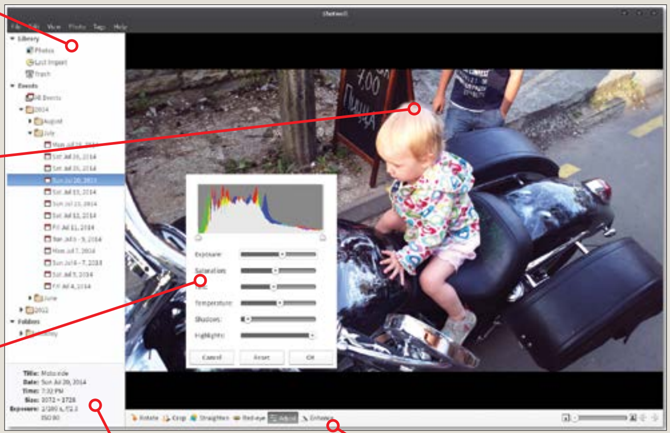


› *Shotwell* è un ottimo programma per l'organizzazione delle foto. In più, dispone di alcuni strumenti per il ritocco

> "Organizza in automatico foto e video tramite le informazioni temporali"

sia grafici come RAW. Naturalmente, Shotwell dispone anche di funzioni più avanzate. Per esempio è possibile organizzare le foto in base ai tag, alle cartelle e al timecode. Inoltre riesce anche a ricercare ed eliminare i file duplicati. L'applicazione consente alcune modifiche basilari, come la rotazione, la regolazione del colore e l'orientamento. In più, niente vieta di visualizzare presentazioni e condividere le immagini con i principali servizi Web come Facebook o Flickr. Nella versione 0.22, gli sviluppatori hanno migliorato il supporto per i metadati. Se volete caricare le vostre foto su Facebook, adesso potete sfruttare anche il supporto per i tag e i titoli. Shotwell 0.22, inoltre, è stato oggetto di un profondo restyling a livello di codice, con il conseguente miglioramento della struttura e la risoluzione di numerosi bug. Non sono poi mancati alcuni miglioramenti all'interfaccia. Per chi usa Ubuntu, è disponibile un PPA (**ppa:yorba/ppa**) che supporta le versioni del programma più moderne. Le altre distribuzioni dovrebbero comunque aver ricevuto Shotwell 0.22, ma è sempre meglio controllare per sicurezza.

L'interfaccia di Shotwell



Barra di navigazione
Da qui è possibile navigare nel disco fisso e nelle cartelle che contengono le immagini.

Immagine in grande
Shotwell può mostrare l'immagine selezionata a tutto schermo (premete **F11** per attivare o disattivare l'opzione).

Regolazioni mirate
Per ritoccare in modo basilare le foto, è disponibile una modalità automatica che rende più veloce l'operazione.

Dati EXIF
In questo angolo è presente il riepilogo completo dei dati EXIF della foto aperta.

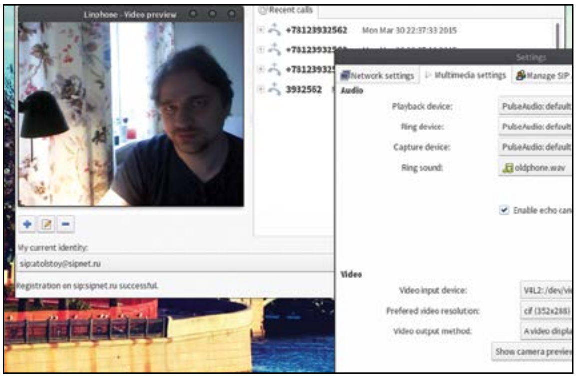
Strumenti di editing
È possibile ruotare, tagliare e raddrizzare le immagini usando la modalità manuale o automatica.

Client SIP

Linphone

Versione: 3.8.5 **Web:** www.linphone.org

La richiesta di applicazioni VoIP è largamente sottovalutata nel mondo Linux. Abbiamo programmi proprietari come Skype, nonché altre buone soluzioni come Tox o Jabber, ma siamo ancora lontani dalla perfezione. **Linphone** è un altro software VoIP/SIP che si distingue per essere una soluzione Open Source multi-piattaforma. Visitando il sito del progetto, si fa fatica a pensare che il tutto si basi su specifiche Open. Tuttavia ci siamo dovuti ricredere non appena abbiamo visitato la pagina per il download. Linphone, infatti, è disponibile per quasi ogni distro desktop principale, nonché per gli OS mobile. Il software è in grado di supportare chiamate audio e video, nonché la messaggistica di testo. C'è anche un servizio *Web-based*, ma anche una potente SKD SIP/VoIP per lo sviluppo di soluzioni di terze parti. Per la maggior parte delle persone, Linphone può essere un valido sostituto di Skype, specialmente se viene usato per le chiamate tra smartphone o computer. Il programma è incluso nei repo predefiniti di Ubuntu e Debian (**sudo apt-get install linphone**). Se poi date un'occhiata a **http://pkgs.org**, troverete un vasto elenco di altre distro che mettono a disposizione Linphone in modo predefinito. Per usare il programma, dovrete prima registrare un account SIP che può comunque essere fornito da qualsiasi servizio a vostra scelta (**@linphone.org**). Una volta effettuato l'accesso, si possono stabilire videochiamate in alta qualità (viene supportata la maggior parte dei codec più moderni). In più, si può sfruttare diverse opzioni come la gestione delle chiamate, l'uso della rubrica, la regolazione della banda e molto altro ancora. La nuova versione 3.8 dispone della modalità a schermo intero.



› *Linphone* **permette di effettuare chiamate video in alta qualità e funziona su ogni desktop in circolazione**

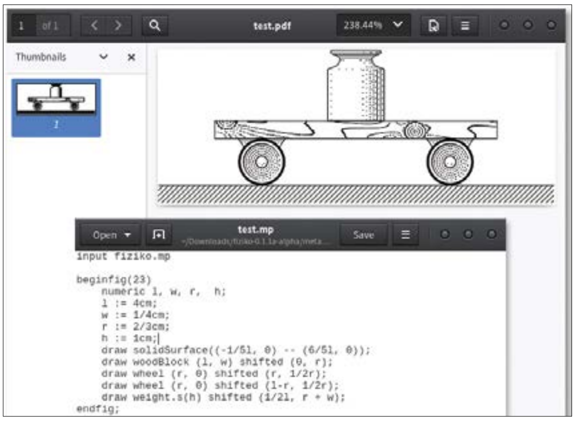
> **"Supporta le chiamate audio e video, nonché la messaggistica di testo"**

Pacchetto MetaPost

Fiziko

Versione: 0.1 **Web:** http://bit.ly/Fiziko

Molte persone preferiscono usare Tex o programmi LaTex al posto di un word processor WYSIWYNG. Tex, infatti, è noto per le sue funzionalità nella progettazione di formule matematiche o chimiche, oltre che per offrire un controllo preciso sulla formattazione del testo. Per anni Tex è stato accompagnato da un linguaggio di programmazione MetaPost e dall'interprete Metafont di Donald Knith. MetaPost eccelle nella produzione di diagrammi in PostScript tramite la sintassi dichiarativa utile per manipolare curve, punti e forme geometriche. **Fiziko** è un pacchetto MetaPost, inizialmente progettato per facilitare la creazione di illustrazioni per libri di testo incentrati sulla fisica. In più, include una serie di macro di "basso livello" che rendono esteticamente gradevoli le illustrazioni in bianco e nero. L'idea alla base del pacchetto è rendere la grafica compatibile con la stampa in alta qualità, migliorando quindi la resa su carta. Fiziko, infatti, imita la xilografia tramite l'uso di tratti sottili che danno un aspetto curato ed esteticamente molto bello alle illustrazioni. Modificando la larghezza del tratto è possibile adattare la grafica anche a stampanti obsolete che non dispongono di un'alta risoluzione. Fiziko, inoltre, dispone di diverse macro che consentono un facile accesso ad alcuni oggetti di base, come curve, blocchi, occhi, pulegge, ruote, carrelli e molto altro ancora. Il tutto, tramite l'inserimento di codice MetaPost facilmente leggibile e pratico da usare per le combinazione complesse. Una volta che avete finito con il vostro file che avrà estensione **.mp**, convertitelo in Postscript, quindi in PDF. Avrete quindi un ottimo documento da usare per ogni esigenza.



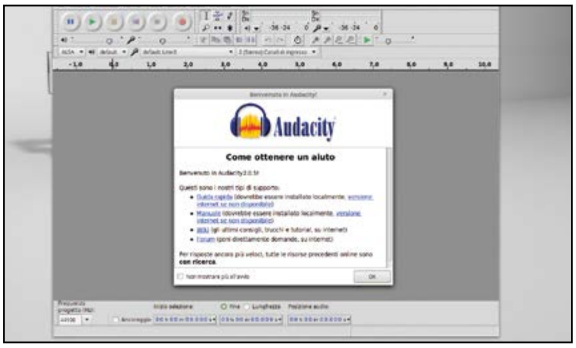
› *Fiziko* **perfeziona la resa delle illustrazioni e le ottimizza per le operazioni di stampa**

> **"Fiziko imita la xilografia tramite l'uso di tratti sottili"**

Editor audio

Audacity

Versione: 2.1 **Web:** http://audacity.sourceforge.net



› Per anni uno dei punti forti di ***Audacity*** **è stata l'interfaccia pratica da usare anche per i principianti**

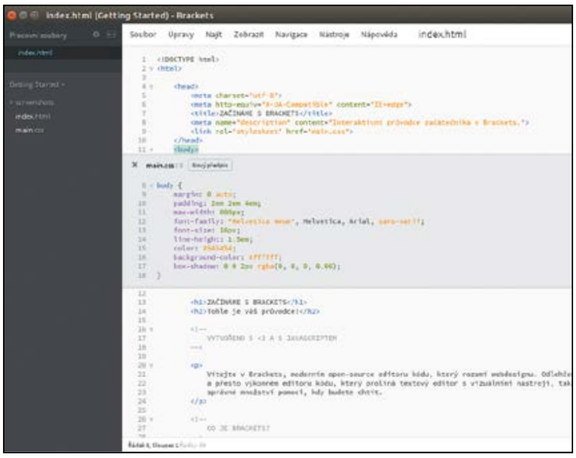
Audacity è uno dei pesi massimi nella gestione dei file audio su Linux. L'ultima versione (2.1) ha visto uno sviluppo di oltre tre anni, ma finalmente è stata rilasciata. Il programma può essere annoverato tra gli editor Open Source multitraccia che consentono sia la modifica sia la registrazione. Il tutto basato su GTK e wxWidgets. Si può registrare il flusso audio da un microfono, nonché registrare la riproduzione da un computer. Tutte le operazioni di input e output vengono effettuate sia attraverso il classico ALSA, sia tramite il moderno sistema di PulseAudio. È inoltre possibile importare i dati audio aprendo semplicemente i file WAV, MP3, FLAC o qualsiasi altro formato supportato (ce ne sono decine). In ogni caso, il programma vi restituisce un grafico su linea temporale, modificabile con vari strumenti integrati. La parte superiore della finestra di Audacity ospita i controlli, nonché la barra degli strumenti di riproduzione. Utilizzando il set base per l'editing è possibile ingrandire e ridurre le dimensioni del grafico, modificare i picchi e disegnare manualmente l'onda sonora. Audacity si rivela un ottimo strumento anche per la miscelazione di più tracce, utile per esempio a creare dei podcast fatti in casa. Uno dei cavalli di battaglia di questo software consiste nella presenza di parecchi filtri incorporati che si trovano nel menu **Effetti**.
Si può sperimentare la modifica del tempo di riproduzione, la dissolvenza, lo stretching, l'aggiunta di eco e molto altro ancora. Audacity si dimostra molto utile anche nella riparazione di registrazioni danneggiate. Dispone di un filtro per la riduzione del rumore, nonché un normalizer più altre funzioni minori che contribuiscono a ridare vita alle vecchie incisioni.

> **"Un programma che non può mancare nei PC di chi ama la musica"**

Editor di testo

Brackets

Versione: 1.2 **Web:** https://github.com/adobe/brackets



› L'interfaccia grafica di ***Brackets*** **è pratica e intuitiva. L'ambiente, infatti, si distingue per la pulizia generale**

Adobe Systems ha iniziato a parlare del suo editor di testo Open Source con licenza MIT a partire dall'estate del 2012. La prima versione stabile di **Brackets** è comparsa nel mese di novembre 2014, ma a quel tempo non riuscì a fare breccia nei cuori degli utenti Linux. Il motivo riguardava soprattutto la mancanza di pacchetti precompilati per la maggior parte delle distro. L'unico disponibile era il PPA per Ubuntu (**ppa:webupd8team/brackets**). Negli ultimi mesi, però, questo gap è stato risolto e adesso il programma può essere utilizzato nella maggior parte delle distribuzioni desktop. L'editor è pensato per il Web Design e lo sviluppo di pagine HTML, ma anche di CSS e JavaScript.
Una volta lanciato, vi troverete in un ambiente molto pulito e ordinato che ricorda molto vagamente Adobe Photoshop. La barra di sinistra mostra la lista degli ultimi file su cui si è lavorato. La parte principale è adibita a editor con relativa barra degli strumenti disposta nella parte superiore. Brackets dispone di diverse caratteristiche interessanti che rendono lo sviluppo Web meno ostico e faticoso. Per esempio, quando si aggiunge il tag **<color>**, si apre un suggerimento automatico che riporta i colori W3C, con tanto di nomi dei campioni. Quando si fa click su un tag in HTML e si preme **Ctrl+E**, vedrete comparire il corrispondente codice CSS. I suggerimenti predefiniti che il programma fornisce supportano la maggior parte dei linguaggi di markup conosciuti, compreso lo standard SVG. Nell'interfaccia compare anche una barra verticale sottile disposta sulla destra che offre due funzioni extra: l'anteprima e la gestione delle estensioni. La prima permette di aprire il codice all'interno di un browser (attualmente solo Chrome), mentre la seconda consente di installare diversi plug-in che forniscono ancora più strumenti al programma.

> **"Brackets dispone di ottimi strumenti per la programmazione Web"**

Browser Web

Netsurf

Versione: 3.3 **Web:** www.netsurf-browser.org



› Con *Netsurf*, la maggior parte delle pagine, anche se non tutte, viene visualizzata correttamente

Mentre la battaglia tra Chrome e Firefox non accenna a diminuire, molti altri ottimi browser di secondo livello vengono oscurati da questi due giganti. Se guardate un po' più là del giardino di casa, infatti, potrete trovare ottime alternative. **Netsurf** è una di queste. Leggero, incredibilmente veloce e ricco di supporto, dispone di parecchie caratteristiche moderne che consentono una navigazione tra le più funzionali che si possano provare. Il programma è molto diverso da qualsiasi altro browser in circolazione. Netsurf è stato progettato nel 2002 e ha iniziato la sua vita operativa come programma per la navigazione dedicato a RISC OS. Nel 2004 ha subito un porting in GTK ed è stato messo a disposizione della comunità Linux. Da allora, il team di sviluppo ha fornito supporto standard per il Web, soprattutto per HTML 4/5 e CSS. Attualmente, ha raggiunto buoni livelli di compatibilità, anche se ad onor del vero c'è ancora tanta strada da fare. Il supporto JavaScript e di molte altre funzioni HTML5 continua a essere carente, ma il resto del browser funziona comunque bene. La funzione di auto-completamento degli URL, così come la gestione di cookie e HTTPS sono ben integrati nel programma. Netsurf, inoltre, occupa davvero poco spazio su disco e fa uso di una quantità di RAM irrisoria. Abbiamo scoperto che richiede solo 15-25 MB. L'ultima versione 3.3 è stata sviluppata per quasi 10 mesi, per poi essere rilasciata ufficialmente nel novembre 2014. Dispone del supporto standard DOM, JavaScript, proxy (diverso da HTTP), temi GTK e maggiori proprietà CSS. Per provare NetSurf dovrete compilarlo da sorgenti. Il processo non è particolarmente difficile, anche grazie alla guida (**http://bit.ly/1lPgxMq**) che documenta ogni passaggio.

> “Netsurf occupa poco spazio e usa una quantità ridicola di RAM”

Client Mail

Geary

Versione: 0.10 **Web:** http://bit.ly/GearyMailClient



› L'interfaccia di *Geary* ricorda molto Thunderbird. In più, dispone di molte funzioni avanzate

Yorba Foundation è stata molto produttiva di recente. **Geary** è un'applicazione per la gestione della posta elettronica che si distingue per un approccio minimalista e funzionale. È elegante e molto veloce. Inoltre è progettata per il desktop Gnome con un stile GTK3. Il software è scritto in Vala e utilizza un database SQLite3 per archiviare le email, nonché una libreria GObject per accedere alle cartelle IMAP. Dopo aver collegato almeno un account di posta, viene creato automaticamente un indice full-text che permette una ricerca veloce e mirata su tutte le caselle email. L'interfaccia di Geary è sviluppata secondo un approccio molto comune: abbiamo tre pannelli che consentono di accedere alle varie funzioni e vedere l'anteprima delle email. Sotto questo profilo, ricorda molto Thunderbird. Sul lato sinistro trovate un elenco degli account, con le relative caselle di posta. Sotto di esse sono presenti gli indici espandibili che mostrano le varie cartelle create. Selezionandone una, il pannello centrale elenca la lista dei messaggi presenti. Basta fare click su un'email per aprirla a schermo intero. Abbiamo scoperto che Geary 0.10 può essere definito senza problemi un client di posta elettronica avanzato che dispone di tutte le funzioni utili alla gestione delle email. Tra queste, è importante citare l'archiviazione che consente di spostare i messaggi ricevuti in cartelle predefinite. Oltre a queste caratteristiche, troviamo anche un editor WYSIWYG per la creazione di messaggi in HTML, nonché un contenitore per questo genere di email. In più, avete la possibilità di creare e gestire in modo veloce e pratico le caselle di posta elettronica più diffuse, come Gmail, Yahoo! Mail e Outlook.com. Le impostazioni per i server più popolari, infatti, sono tutte memorizzate nel programma. Infine, si dispone del supporto completo per SSL e STARTTLS. Geary è molto semplice da installare su Ubuntu. Dispone di un proprio PPA (**ppa:yorba/ppa**) e in più è disponibile anche per Arch e molte altre distro Linux.

> “Geary 0.10 può essere considerato un client di posta avanzato”

Giochi Applicazioni per il tempo libero

Gioco da tavolo

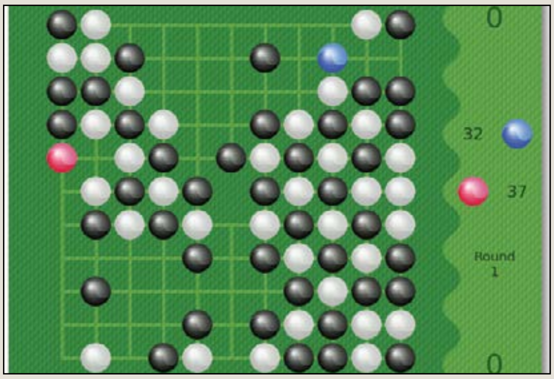
Pasang Emas

Versione: 5.0.0 **Web:** http://bit.ly/PasangEmas

Pasang Emas è un gioco da tavolo tipico del Brunei. La partita è basata sulla cattura delle pedine dell'avversario, il tutto senza rimanere bloccati. Le regole, comunque, sono piuttosto complesse e vale la pena perdere un po' di tempo per studiarle. Il gioco inizia con l'immissione delle pedine su un modello simmetrico. Intorno al centro della tavola deve poi rimanere una casella vuota. I giocatori possono creare vari passaggi scegliendo una fila verticale di cinque pedine. Quando il primo giocatore crea una riga, il secondo può scegliere qualsiasi linea tranne la corrispondente fila simmetrica su un fianco. Come detto, spiegare le regole può essere piuttosto complicato, quindi vi rimandiamo alla documentazione che trovate sul sito ufficiale. Pasang Emas può essere personalizzato. Infatti, sempre sul sito, sono disponibili vari temi per la scacchiera che può assumere l'aspetto di un muro di mattoni o di una tavola di legno. Per l'installazione bisogna utilizzare i seguenti passaggi dopo aver scaricato il file **pasang-emas-theme-1.0.tar.bz2**:

```
$ tar -xjf pasang-emas-themes
1.0.tar.bz2
```

> “Le regole del gioco sono complesse e vanno studiate con attenzione”



› Le tavole di *Pasang Emas* possono essere modificate con vari temi scaricabili dal sito ufficiale

```
$ cd pasang-emas-themes-1.0
$ cp -r kaca $HOME/.pasang-
emas/themes/
$ cp -r content $HOME/.
pasang-emas/themes/.
```

Pasang Emas è disponibile per tutte le piattaforme, ma è stato maggiormente provato sulle distro Debian. Con l'ultima versione diventa anche multi-piattaforma, riuscendo a funzionare anche su Mac.

Gioco di strategia

FreeCol

Versione: 0.11.5 **Web:** http://bit.ly/FreeColGame

Come suggerisce il nome, **FreeCol** è una fork Open Source del famoso titolo di strategia uscito nel 1994. Il gioco inizia nel 1492, anno della scoperta dell'America. Il vostro scopo sarà proprio colonizzare il nuovo mondo. Si inizia con una sola nave e due coloni in mezzo all'Oceano Atlantico. Dovrete quindi navigare tra le onde fino a trovare l'agognata terra. Proprio come in Civilization (un altro gioco di strategia molto popolare), è necessario costruire una nazione potente. Nella prima parte, però, sarete anche in grado di inviare navi in Europa per rifornirvi di vettovaglie e trasportare nuovi coloni. Proprio a questo proposito, riuscire a rimpolpare le fila delle vostre genti in terra straniera è un'operazione molto importante. Potrete pagare i coloni per giungere nel nuovo Mondo, oppure promettere la libertà di religione per invogliarli a venire di loro spontanea volontà. Più coloni avete a disposizione e maggiori saranno le possibilità e la velocità di sviluppo. FreeCol migliora comunque il gioco originale con diversi accorgimenti. Uno di questi è la possibilità di sfruttarlo in modalità multi-piattaforma. Su Linux, per esempio, è basato su Java ed è distribuito con il programma d'installazione JAR.

> “FreeCol migliora il gioco originale nato nel lontano 1994”



› Potete invogliare i coloni a trasferirsi dall'Europa all'America con la promessa della libertà di culto

Per installarlo, è sufficiente assicurarsi di avere OpenJDK, quindi scrivere il comando:

```
java -jar FreeCol-0.11.3-installer.jar
```

A questo punto non vi resterà altro da fare che seguire le istruzioni a schermo. Per impostazione predefinita, il gioco viene archiviato nella directory **~/freecol**, dove troverete anche lo script di avvio.

Software per la virtualizzazione

VirtualBox

Versione: 5.0.0 beta1 **Web:** http://bit.ly/VirtualBox5B1



› Le ultime Guest Addition per *VirtualBox* migliorano sensibilmente l'esperienza d'uso del programma

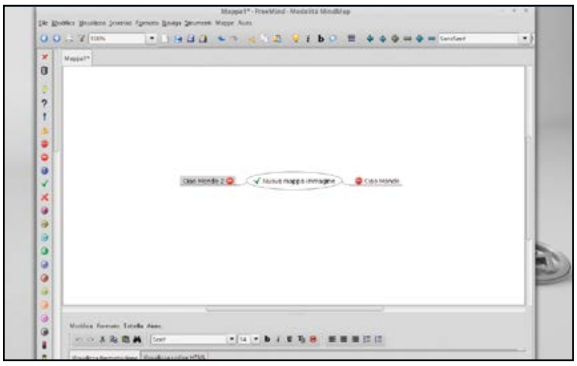
Sono passati parecchi mesi da quando **VirtualBox** si è presentato al grande pubblico con una buona serie di innovazioni. Molti appassionati di questo software si sono perfino detti preoccupati per lo stallo che l'applicazione sembrava aver subito da quando Oracle lo ha rilevato da Sun Microsystems nel 2010. Sebbene il software sia gestito da Oracle, comunque, rimane un progetto Open Source che purtroppo viene sviluppato secondariamente rispetto ad altri. Adesso è però venuto il momento che tutti aspettavano. La versione beta 5.0 di VirtualBox porta con sé una serie di cambiamenti degni di nota, tra cui l'ottimizzazione per la virtualizzazione di ambienti Windows e Linux. Questa Beta, inoltre, dispone anche di diverse estensioni per i set di istruzioni delle macchine virtuali guest. Aggiunge il supporto per le USB 3.0 nel controller XHCI, nonché la condivisione bidirezionale degli appunti. Da non dimenticare poi la nuova gestione della crittografia. Ci sono anche diversi miglioramenti a livello di interfaccia grafica, ma anche per quanto riguarda il codice che è stato ottimizzato. Adesso, infatti, possiamo sfruttare un programma privo di bug, veloce e funzionale su ogni distro. La caratteristica più interessante di questa Beta è però il supporto per la para-virtualizzazione. A tal proposito, vengono offerte diverse interfacce adatte allo scopo, come Legacy, Minimal, Hyper-V (per Windows) e KVM (per Linux). La para-virtualizzazione fornisce strumenti che permettono a un sistema virtualizzato di accedere a periferiche hardware presenti nel computer host. Il tutto si traduce quindi in un sensibile aumento delle prestazioni. Nei nostri test, per esempio, la nuova versione di VirtualBox ha dimostrato un incremento di velocità notevole, così come un accesso meno intensivo al processore.

> “La più grande novità della Beta 5.0 è la para virtualizzazione”

Software per mappe mentali

FreeMind

Versione: 1.0.1 **Web:** http://freemind.sourceforge.

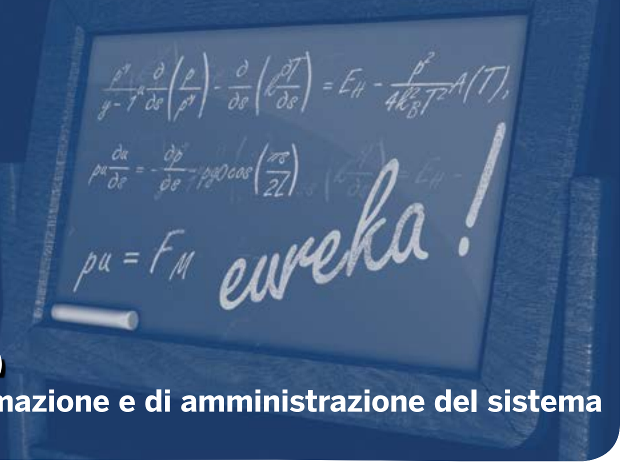


› Con *FreeMind* potete creare delle mappe logiche molto flessibili in tutti i loro aspetti

FreeMind è un software specializzato per la creazione di mappe mentali, utili per la gestione dei progetti complessi. Come altri programmi simili, FreeMind permette a un utente di modificare un insieme gerarchico di idee che ruotano intorno a un concetto principale. Visto che i vari rami di un'idea circolano intorno al fulcro, lo sviluppo grafico non è lineare.
Il progetto dispone di diversi download. Ciò nonostante, consigliamo di usare l'archivio **freemind-bin-max.1.0.1.zip**, scompattarlo ed eseguire **./freemind.sh** all'interno del terminale.
La finestra principale dispone di un ampio spazio, nonché di piccole icone su entrambe le barre degli strumenti orizzontali e verticali. La maggior parte delle interazioni, comunque, potrete gestirle tramite svariate combinazioni di tasti che possono essere apprese dalla documentazione del programma. Per cominciare, per esempio, è possibile iniziare con una mappa che contiene il concetto principale intorno a cui devono ruotare tutte le altre idee. Premete **Ins** per creare il primo nodo figlio, quindi **Invio** per generare un nodo fratello, infine **Shift**+**Invio** per un nuovo ramo figlio. Facendo click con il tasto destro su un nodo, compare un menu contestuale in cui si può agire su diverse funzioni: personalizzare il nodo, aggiungere una o più icone, gestire il font per l'etichetta, inserire il disegno di una nuvola, un'immagine o un collegamento ipertestuale.
L'immagine campione mostra la varietà di opzioni utilizzabili per creare e personalizzare una mappa colorata e complessa. Per accedere al testo di ciascun nodo, occorre premere **Ctrl**+**F**. Se poi decidete di riorganizzare alcuni dei vostri obiettivi, vi consigliamo di posizionare i nodi in base alle esigenze. La documentazione è molto varia e ben realizzata. Potete accedervi in modo ricorsivo tramite il formato FreeMind basato su Java che richiederà comunque al browser l'uso del plug-in **IcedTea**. LXP

> “Grazie alle opzioni di personalizzazione si creano mappe colorate”


PRENOTA QUI
LA TUA COPIA
www.spreastore.it/minecraft
È in edicola
COME FARE TUTTO IN
MINECRAFT
MINECRAFT
CORSO DI SOPRAVVIVENZA
Le tecniche avanzate per superare le insidie della notte!
LAVA, CREEPER, NETHER
Conoscere il mondo di gioco per dominarlo con i consigli giusti!
I FERRI DEL MESTIERE
Uno studio approfondito sulle armi e sugli attrezzi!
I BIOMI ■ I GADGET ■ I TRUCCHI ■ L'HARDCORE ■ LE POZIONI ■ LE ARMATURE
100% INDIPENDENTE E NON UFFICIALE
Sprea
Sprea editori



Tutorial

I nostri esperti offrono i loro consigli di programmazione e di amministrazione del sistema

LA GUIDA DI RIFERIMENTO

Esiste sempre qualcosa di nuovo da imparare in campo informatico, soprattutto in un mondo dinamico come quello di Linux e dell'Open Source. Ogni numero di Linux Pro presenta una sezione dedicata a tutorial realizzati da esperti in moltissimi settori: programmazione, sicurezza, amministrazione di sistema, networking. Troverete informazioni utili sia che siate dei veterani di Linux sia degli utenti alle prime armi. Studieremo con cura anche le applicazioni più diffuse sia in ambito lavorativo che desktop. Il nostro scopo è quello di fornire in ogni numero il giusto mix di argomenti, ma se avete suggerimenti su temi particolari che vorreste vedere trattati, scriveteci via e-mail all'indirizzo tutorial@linuxpro.it

COME RAPPRESENTIAMO LE LINEE DI CODICE

Si presenta spesso la necessità di riportare le linee di codice di un programma. Per favorirne la lettura evidenzieremo le singole linee in questo modo:

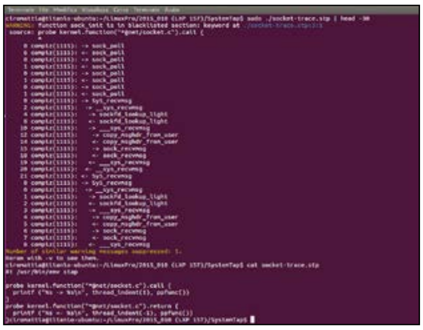
```
begin
  mniWordWrap.Checked := not
end
```

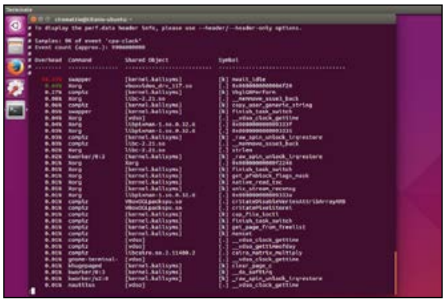
Quando una riga di codice supera la lunghezza della colonna la riporteremo su più righe utilizzando la notazione seguente:

```
printf("Vi preghiamo di inserire
una password.");
```

TUTORIAL





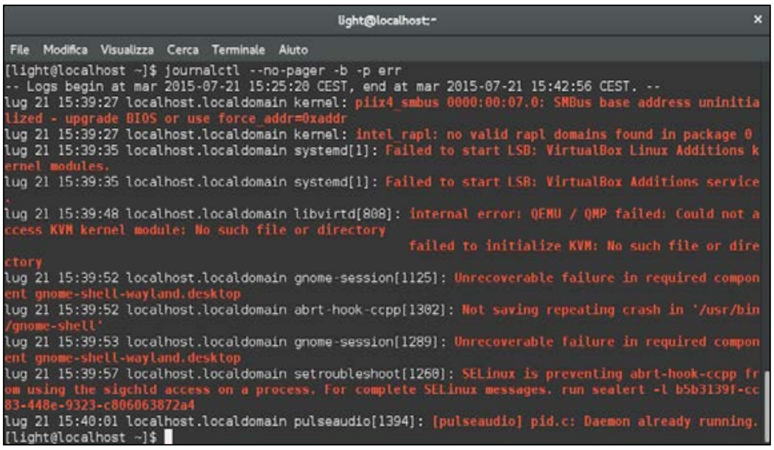


ACCADEMIA DEL CODICE

Un tuffo dentro a journalctl

Esploriamo il sistema di log di **Systemd** chiamato journal e tentiamo di sviscerare per voi la maggior parte delle sue funzionalità

Dato che molte distribuzioni stanno passando a Systemd, il modo di gestire i log sta cambiando e noi dobbiamo abituarci. La novità principale è la comparsa del sistema di log di Systemd. Ma cos'è questo Journal? In cosa differisce da syslog? Come lo possiamo utilizzare? Possiamo continuare a utilizzare quello che usavamo prima se vogliamo? Il journal di Systemd raccoglie e salva i dati dei log da molte fonti: i log del kernel, lo standard output, gli errori dai servizi e molto altro.
Il journal salva più informazioni su ogni messaggio rispetto ai vecchi demoni syslog e salva tutto in un formato strutturato e indicizzato che permette una maggior facilità di ricerca. Il comando principale per leggere i dati da journal è **journalctl**. Eseguendolo senza argomenti vi mostrerà un insieme di contenuti simile al sistema di log tradizionale. Ci sono però alcune differenze: la prima è che journalctl utilizza automaticamente un sistema di paging, di solito **less**, per visualizzare le informazioni. Potete notare inoltre che risale di parecchio tempo, possibilmente fino alla prima installazione. Questo perché il journal non si basa su **logrotate** per eliminare i vecchi dati ma li gestisce in autonomia comprimendoli e mantenendoli comunque ricercabili. I file vengono mantenuti finché la loro dimensione non supera la soglia impostata in **/etc/systemd/journald.conf**.



› **Questo output è tutto rosso perché abbiamo utilizzato l'opzione *-p err* per vedere soltanto gli errori**

Mantenersi aggiornati

Anche se può essere utile avere una lunga storia di messaggi di sistema, potreste notare un po' di ritardo quando eseguite **journalctl**. Fortunatamente, journalctl ha diversi modi per limitare le ricerche e, siccome il journal è indicizzato, lo fa molto velocemente. Per esempio:

```
journalctl -b
```

mostra i messaggi a partire dall'ultimo riavvio. Potete utilizzare anche opzioni come:

```
journalctl --since today
journalctl --since 09:30 --until 11:00
```

La prima mostra i dati dalla mezzanotte, la seconda quelli compresi nell'intervallo temporale impostato. Potete utilizzare le date e i timestamp così come delle stringhe come today (oggi) o yesterday (ieri) come argomenti per **--since** e **--until**.
Il journal di **Systemd** salva molti dati, più dello standard syslog, quindi potreste trovarvi sopraffatti dalle informazioni che vi ritorna. Molte informazioni salvate in qualsiasi sistema di log sono superflue, quindi diamo un'occhiata soltanto alle cose più importanti:

```
journalctl -b -p err
```

Questo comando mostra soltanto i messaggi successivi all'ultimo riavvio, con una severità pari all'errore o superiore.
A volte può far comodo vedere soltanto i dati più recenti o attuali:

```
journalctl -n 10
journalctl -f
```

La prima riga mostra gli ultimi 10 inserimenti, mentre **-f** segue il journal, mostrando i dati mano a mano che vengono aggiunti. In pratica è come fare **tail -f /var/log/messages** ma senza che il buffer di scrittura si metta in mezzo.

Filtrare

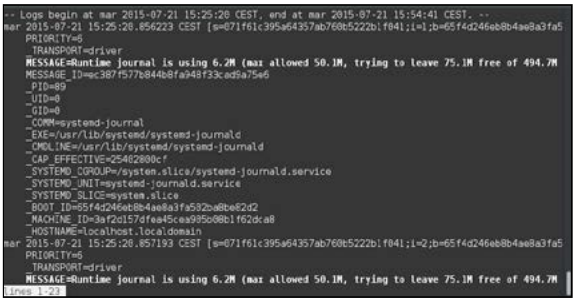
Fin qui, l'output di **journalctl** non si è distinto molto da quello dei normali file presenti in **/var/log/messages**, quindi, dove sono tutte queste informazioni aggiuntive? Per vederle vi basterà lanciare questo comando:

```
journalctl -o verbose
```

Come potrete vedere con questo comando otterrete moltissime informazioni aggiuntive, e la cosa migliore è che potete fare ricerche nel journal utilizzando ognuna di esse. Non c'è più bisogno di utilizzare **grep** per filtrare i log. Assumiamo che oggi abbiate avuto un problema con una condivisione **Samba**, potete cercare di scoprire cos'è accaduto con:

Mantenere la vecchia rotta

Esistono un paio di ragioni per cui potreste voler continuare a utilizzare i vecchi file di log in ***/var/log***. Se utilizzate qualcosa tipo **logcheck** o **logwatch** per controllare i vostri log di sistema, siete costretti a fornirgli un file di log standard come ***/var/log/messages***. Il modo più semplice per farlo è quello di installare un logger di sistema classico, come **syslog-ng**. Non ci sono problemi nel farlo, il sistema sarà felice di salvare i dati sia sul journal sia su file, comunque speriamo che un po' alla volta tutti i programmi si adatteranno al nuovo sistema di log. L'altra ragione è la comodità. Mentre vi state abituando al nuovo modo di accedere ai messaggi del vostro sistema, potete mantenere il vecchio sistema in esecuzione. Certo, è lecito chiedersi se sia una buona idea, Avere la possibilità di utilizzare **syslog-ng** tutte le volte che vogliamo difficilmente ci sproneràa imparare a utilizzare **journalctl**. Qualunque sia la ragione per cui potreste voler eseguire un logger di sistema differente, se deciderete di farlo ricordatevi di installare anche **logrotate**. A differenza del journal di **Systemd**, la maggior parte dei demoni syslog non si occupano di ruotare i vecchi dati, quindi dovrete utilizzare **logrotate** per far sì che non vi riempiano tutto l'hard disk.



› **Eseguendo *journactl* con l'opzione *-o verbose* possiamo vedere tutti i dati aggiuntivi. Ognuno di questi dati può essere utilizzato per fare ricerche**

```
journalctl --since today _COMM=smbd
```

_COMM esegue la ricerca nel nome del comando, potete utilizzare **_EXE** per cercare sull'intero percorso. In effetti jounalctl cerca di processare gli argomenti che gli mandate in modo intelligente, quindi, se gli date il percorso di un eseguibile lui lo tratterà come filtro per **_EXE**. Se gli passate il percorso di un device, vi mostrerà i messaggi kernel legati a quel device, in pratica vi basterà utilizzare delle chiamate come queste:

```
journalctl /usr/sbin/crond
journalctl /dev/sda
```

Esistono diversi campi su cui potete fare ricerche, potete trovare l'elenco completo nella man page systemd.journal-fields ma potete vederli anche utilizzando l'opzione di output verbose. Ovviamente, nessuno vuole dover guardare la documentazione ogni volta che ha bisogno di fare una ricerca e fortunatamente non ce n'è bisogno. L'autocompletamento di **bash** può aiutarvi. Volete cercare nel campo unit di Systemd ma non ricordate il nome esatto? Provate:

```
journalctl _SYS<TAB>
```

Potete utilizzare l'autocompletamento anche sui valori, anche se può rivelarsi abbastanza lento la prima volta. Inoltre potete filtrare su più di un campo o più volte sullo stesso campo, per esempio:

```
journalctl _SYSTEMD_UNIT=crond.service _SYSTEMD_
UNIT=collectd.service
```

In questo caso journalctl mostrerà sia i valori che corrispondono al primo filtro che quelli che corrispondono al secondo. Se invece mettete filtri su campi diversi mostrerà soltanto i valori che corrispondono a entrambi i filtri a meno che non li separiate con **+**.

Configurazione di journal

Abbiamo visto che journalctl passa l'output al vostro pager predefinito. Esistono due modi per evitare che succeda. Potete utilizzare l'opzione **--no-pager** oppure potete inviare l'output da qualche altra parte invece che sul terminale, ridirezionandolo su un file o mettendolo in pipe con un altro programma. In certi casi può ancora fare comodo filtrare l'output utilizzando **grep** o **awk**. Il journal è mantenuto in memoria, per garantire un accesso più rapido, ma può anche essere scritto su disco. Se la directory ***/var/log/journal*** esiste e l'utente systemd-journal ha il permesso di scriverci, i vostri log verranno salvati anche lì. Dato che il journal comunque resta anche in memoria, journald salva tutto quello che accade da quando Systemd parte, e poi salva sul disco quando la directory di **journal** diventa disponibile. Questo significa che gestisce anche le informazioni iniziali dell'avvio e i messaggi del kernel, a differenza dei gestori di log tradizionali che iniziano a salvare dati soltanto quando il servizio syslog è partito. Come già detto, journald si occupa anche della rotazione e della compressione dei vecchi log, ma invece di utilizzare la strategia di **logrotate**, utilizza lo spazio disponibile sul disco per determinare quanti dati tenere. Di default è limitato in modo da non utilizzare mai oltre il 10% dello spazio totale sul filesystem e di lasciare sempre almeno il 15% di spazio libero. Se uno di questi limiti viene superato i dati più vecchi vengono eliminati. Queste soglie possono essere modificate cambiando i valori di SystemMaxUse e SystemKeepFree nel file ***/etc/systemd/journald.conf***. In questo file sono presenti molte altre opzioni, come la possibilità di comprimere i dati o di inoltrare i messaggi a syslog (date un'occhiata al box **Mantenere la vecchia rotta** per vedere come farlo e perché può essere una bella idea).

Log remoti

Potete, ovviamente, fare SSH su un computer per usare journalctl, ma potete anche abilitare il servizio systemd-journal-gatewayd.service per rendere i log disponibili tramite HTTP (di default sulla porta 19531, è disponibile anche in HTTPS). Anche se vedere i log in un browser Web può sembrare un po' limitato, potete utilizzare anche systemd-journal-remote per prendere i dati del journal di quel sistema e inserirli in un journal separato all'interno della macchina locale. Tenete presente che journalctl unisce tutti i dati di tutti i journal presenti nel computer, quindi vi mostrerà sia gli eventi locali che quelli remoti. Potete utilizzare il filtro _HOSTNAME per visualizzare quelli di un sistema specifico. Oltre al journal di sistema, Systemd mantiene anche dei journal per gli utenti collegati. Soltanto i superuser, e i membri dei gruppi systemd-journal, adm e wheel possono leggere i journal di sistema. Il comportamento predefinito di journalctl è quello di mostrare i dati di tutti i journal che l'utente è in grado di leggere. Per visualizzare soltanto l'output dei journal di sistema o dell'utente potete utilizzare gli switch **--system** o **--user**. LXP

Tracciare le performance

Lo staff di Linux Pro porrà domande sulle performance del sistema, e userà l'utility da riga di comando **sysstat** per cercare di trovare una risposta

Per prima cosa poniamo una domanda di tipo comportamentale di quelle utilizzate durante i colloqui lavorativi per gli amministratori di sistema: "Ricevete lamentele dalla comunità di utilizzatori perché l'applicazione database che state mantenendo è lenta. Descrivete come risolvereste il problema". Questo tutorial vi aiuterà a trovare una risposta a questa domanda (e forse vi aiuterà a trovare lavoro). In realtà andremo a sondare soltanto una parte della risposta (quindi forse dovrete studiare ancora un po' per quel lavoro). In ogni caso vedremo come analizzare e risolvere i problemi della CPU, della memoria e di I/O sfruttando le misurazioni offerte dal kernel di Linux, e lo faremo su un sistema CentOS. Sysstat è una raccolta di tool che esamina i contatori del kernel. Capire i contatori permette all'amministratore del sistema di monitorare le performance. L'obiettivo è quello di cercare di individuare i problemi di performance utilizzando i contatori del kernel.

Preparazione

Il kernel di Linux offre dei contatori che danno indicazioni sull'attività del sistema. I contatori registrano dati sull'I/O, come la velocità di trasferimento dei dischi, l'uso della memoria di paging, i dati scambiati sulle reti, l'utilizzo dello spazio di swap e molto altro. I dati legati alla CPU sono basati sul suo utilizzo globale o diviso per ogni processo; il pacchetto sysstat offre la capacità di produrre report utilizzando tutti questi contatori. Stiamo utilizzando CentoOS ma l'utility non è un'esclusiva delle distribuzioni basate su Red Hat, potete trovarla anche sulle distribuzioni derivate da Debian.
Per installare questa utility in Debian utilizzate:

```
sudo apt-get install sysstat
```

Invece per Red Hat utilizzate:

```
sudo yum install sysstat
```

oppure

```
sudo rpm -Uvh sysstat-11.0.0-3.fc21.x86_64.rpm
```

Potete vedere l'elenco delle utility contenute in sysstat con il comando **rpm -ql sysstat**:

» **/usr/bin/sar** Raccoglie, mostra, o salva le attività di sistema;
» **/usr/lib64/sa/sadc** Raccoglie dati sull'attività di sistema;
» **/usr/lib64/sa/sa1** Raccoglie e salva dati binari nei file quotidiani che contengono i dati sull'attività di sistema e si trovano nella cartella **/var/log/sa**;
» **/usr/lib64/sa/sa2** Crea dei resoconti leggibili dai file della cartella **/var/log/sa**;
» **/usr/bin/sadf** Mostra i dati raccolti da **sar** in formati diversi (es: CSV, XML);
» **/usr/bin/iostat** Mostra le statistiche su CPU e I/O per i device e le partizioni;
» **/usr/bin/mpstat** Mostra le statistiche sulla CPU per ogni processo;
» **/usr/bin/pidstat** Mostra le statistiche su un task in base al suo pid;
» **/usr/bin/nfsiostat-sysstat** Mostra le statistiche I/O per NFS;
» **/usr/bin/cifsiostat** Un resoconto che consiste in una riga di header CIFS seguita da una riga di statistiche per ogni filesystem CIFS montato.

L'installazione del pacchetto software fa alcune modifiche al crontab per stabilire un ciclo di reperimento dati quotidiano.

```
cat /etc/cron.d/sysstat
# Run system activity accounting tool minutes
*/10 * * * * root /usr/lib64/sa/sa1 1 1
# 0 * * * * root /usr/lib64/sa/sa1 600 6 &
# Generate a daily summary of process accounting at 23:53
53 23 * * * root /usr/lib64/sa/sa2 –A
```

Abbiamo cercato di non dilungarci troppo nell'installazione o nelle spiegazioni per avere più spazio per concentrarci su tre utility: **sar**, **sa1** e **sa2**. Se volete avere maggiori dettagli vi suggeriamo questo blog **http://sebastien.godard.pagesperso-orange.fr/** (in inglese).

Tip

Per trovare i file delle utility di **sysstat** su di un sistema a 32-bit, togliete il 64 dai percorsi delle directory.

Lavorare con sar

L'utility centrale del pacchetto sysstat è il comando **sar**. È un'utility che serve a fare resoconti sull'attività di sistema. Il comando legge i contatori dal kernel Linux e li mostra nello standard output (per esempio sullo schermo). Il comando **sar** supporta parametri e opzioni. Le opzioni **sar {opzione} <intervallo> <iterazioni>** sono utilizzate per raccogliere informazioni su attività di sistema specifiche. I parametri intervallo e iterazioni che seguono l'opzione sono utilizzati dal comando per definire quando e quante volte mostrare i dati. La struttura di comandi **sar** che segue genera un report di 60 secondi con diversi intervalli temporali e numero di iterazioni. L'opzione **-A** fa sì che vengano considerate tutte le attività:

```
/usr/bin/sar -A 5 12
/usr/bin/sar -A 10 6
/usr/bin/sar -A 20 3
```

Senza utenti

Tutti gli amministratori di sistema vi potranno confessare che il loro sistema funzionerebbe molto meglio se non ci fossero utenti a utilizzarlo, ma in quel caso non avrebbero nemmeno un lavoro. Gli utenti offrono dettagli utili relativi ai problemi. I problemi possono trovarsi in quattro aree, oltre agli utenti abbiamo: la parte relativa alla rete che comprende tutti i componenti che vanno dall'host fino al server che offre l'applicazione; il software, i cui problemi possono essere nel sistema operativo, nei database o nelle applicazioni; l'hardware: la CPU, la memoria oppure l'input output I/O. Ottenere informazioni utili alla risoluzione dei problemi dai report degli utenti può essere un'impresa ardua. Sapere quando il problema si presenta è un'informazione utile per accelerare i tempi. Il kernel offre diversi contatori per tracciare l'attività della CPU, l'utilizzo della memoria e le statistiche sull'I/O relative al momento. Sapere il momento in cui il problema si presenta e conoscere i relativi contatori del kernel può aiutare un amministratore di sistema a capire se il problema può essere legato all'hardware.

Nel primo caso viene generato un report ogni 5 secondi per 12 volte (per un totale di 60 secondi), nel secondo 6 report a distanza di 10 secondi (60 secondi) e nel terzo 3 report, uno ogni 20 secondi (sempre 60 secondi). Utilizzare un'opzione o una combinazione delle 18 opzioni disponibili permette di visualizzare i dati delle attività desiderate.
La man page del comando **sar** contiene una descrizione per ogni campo disponibile. Potete leggerla anche online andando a questo link **http://linux.die.net/man/1/sar** se lo trovate più comodo. Il comando **sa1** raccoglie i contatori del kernel e li salva in un file binario quotidiano **/var/log/sa/sa#**. L'altro comando di sysstat **sa2**, traduce i file binari in ASCII leggibili che vengono salvati in **/var/log/sa/sar#**. Le opzioni e i parametri del comando **sar** sono utilizzabili anche con il comando **sa2**. I report ASCII possono essere generati per attività specifiche utilizzando le stesse opzioni utilizzate con il comando **sar**. Il # che abbiamo messo dopo il nome del file viene sostituito automaticamente dal comando con un numero che corrisponde al giorno del mese.
La configurazione di crontab che abbiamo visto fa sì che la routine di **sa1** venga eseguita giornalmente. Crontab lancia anche il comando **sa2** per tradurre i dati binari raccolti da sa1 in un report ASCII. Queste attività di cron ritornano un elenco di attività quotidiane che un amministratore di sistema può utilizzare per controllare eventuali problemi di performance. Come potete notare i parametri <intervallo> e <iterazioni> impostati nel crontab non sono molto granulari. Potete notare anche un'azione commentata, rimuovendo il commento potrete avere una quantità maggiore di informazioni se ne avete bisogno. Eseguendo l'utility **sysstat** tramite cron possiamo raccogliere e salvare informazioni con regolarità. Se si verifica un problema, i dati salvati possono essere analizzati per risolvere il contrattempo, e l'utility **sa1** è un ottimo modo per catturare informazioni durante le fasi di sviluppo. Supponiamo che un amministratore stia testando una nuova routine di query su database. Eseguire l'utility **sa1** durante il test sul database fa sì che venga generato un report con tutti i dati necessari a capire se l'esecuzione del codice sul database ha generato qualche problema. I comandi che andremo a vedere ora servono a creare un report con i dati di dieci minuti. La prima riga di comando inserisce un record falso nel file per assicurarci di resettare i dati:

```
usr/lib64/sa/sa1 --boot
```

Questo comando crea una lavagna pulita per fare i test, ma ricordate che se **sa1** è in esecuzione da crontab, potrebbe avere dai dati in attesa di essere scritti; per questo bisogna disattivare il reperimento dei dati da parte di cron durante i test per assicurarci che i dati siano integri. Con la seconda riga eseguiamo un campionamento ogni 10 secondi per 60 volte quindi 10 minuti di dati:

```
usr/lib/64/sa/sa1 10 60
```

»

Tip

Raccogliere dati dai contatori del kernel consuma risorse. È buona norma mantenere bassa la quantità di dati recuperata e aumentarla solo in caso di necessità.

Interfacce di rete

Tempo	Interfaccia	rx pacchetti/s	tx pacchetti/s	rx kB/s	tx kB/s	rx mcst/s
12:46:09	eth0	9.02	2.81	2.12	0.78	0.6
12:46:14	eth0	3509.18	1758.08	5190.03	122.25	0.2

Trovate la tabella completa all'indirizzo **http://pastebin.com/RzUtbXzF**

Tempo	Interfaccia	rx errori/s	tx errori/s	coll/s	rx drop/s	rx frame/s
12:46:09	eth0	0	0	0	0	0
12:46:14	eth0	0	0	0	0	0

Tempo	Interfaccia	rx pacchetti/s	tx pacchetti/s	rx kB/s	tx kB/s	rx mcst/s
13:38:54	eth0	6314.95	3003.23	9351.52	210.38	0.4
13:38:59	eth0	6400.6	2972.95	9480.99	208.45	0.4

Tempo	Interfaccia	rx errori/s	tx errori/s	collisioni/s	rx persi/s	rx frame/s
13:38:59	eth0	0	0	1545.29	12.22	0
13:39:04	eth0	0	0	1512.05	7.63	0

» La terza riga crea un file ASCII leggibile, mentre la quarta mostra i contenuti nello standard out. Il nome del file ha come suffisso il giorno del mese.

```
usr/lib64/sa/sa2 -A
less /var/log/sa/sar#
```

Come potete notare abbiamo passato al comando **sa2** l'opzione **-A**. Ovviamente potete cambiarla con qualsiasi opzione preferiate per ottenere un file ASCII con informazioni più specifiche. Se avete giocato a calcio, sapete che se l'allenatore dice di fare qualcosa, questo non significa che quella cosa stia per accadere. Quindi andiamo a verificare che i dati siano stati veramente raccolti e cerchiamo di capire quali dati hanno superato il limite. Senza un po' di esperienza può essere difficile interpretare questi file. Controllare contatori singoli può essere un buon modo per restringere l'ambito del problema. Capire come i contatori interagiscono tra loro vi permetterà di valutare l'intero sistema anziché soltanto una parte. Nella prossima sezione useremo le utility di sysstat per catturare dati mentre sono in esecuzione script da riga di comando progettati appositamente per impattare sulle performance del sistema. Possiamo utilizzare i dati raccolti per analizzare e capire meglio cosa succede al sistema.

Gli script

Uno script è studiato per aumentare il carico della CPU. Questo script massimizza l'utilizzo di tutti e quattro i core disponibili nel nostro sistema di test. Un altro script consuma porzioni di memoria disponibile finché non la riempie tutta, che è un po' come vedere lo squalo del famoso film mangiare bagnanti uno dopo l'altro. I dati che vedrete sono presi da un sistema reale. I contatori del kernel sono stati catturati tramite interfaccia di rete dal sistema manomesso. I dati sono stati presi e visualizzati utilizzando i quattro comandi che abbiamo usato prima, e non c'è bisogno di dire che è consigliabile che

Scenari CPU e Memoria

Tempo	CPU	%usr	%sys	%iowait	%soft	%idle
12:51:20	all	0.05	0.2	0	0	99.75
12:51:20	0	0	0	0	0	100
12:51:20	1	0	0	0	0	100
12:51:20	2	0.2	0.8	0	0	99
12:51:20	3	0	0	0	0	100
12:51:25	all	0.05	28.76	0	0	71.19
12:51:25	0	0	38.6	0	0	61.4
12:51:25	1	0	18.64	0	0	81.36
12:51:25	2	0.2	19.14	0	0	80.65
12:51:25	3	0	38.6	0	0	61.4

Reset dei dati (http://pastebin.com/f4PNYSVA)

Tempo	CPU	%usr	%sys	%iowait	%soft	%idle
09:35:22	tutto	0.57	9.34	19.81	0.6	69.68
09:35:22	0	0.4	9.76	35.96	0.94	52.94
09:35:22	1	0.27	7.56	9.02	0.93	82.23
09:35:22	2	1.73	19.65	34.26	0.53	43.82
09:35:22	3	0	0.53	0	0	99.47
09:35:25	tutto	0.3	5.69	49.11	0.44	44.46
09:35:25	0	0.59	7.12	81.31	0.89	10.09
09:35:25	1	0.3	6.23	25.22	0.3	67.95
09:35:25	2	0.29	9.41	56.76	0.59	32.94
09:35:25	3	0	0	33.33	0	66.67

Statistiche sulla memoria (http://pastebin.com/cfa8UPuB)

Time	kb mem libera	kb mem usata	% mem usata	kb cache	kb commit	% commit
09:35:14	1689104	230812	12.02	13600	150032	2.48
09:35:22	59576	1860340	96.9	10968	2323400	38.41
...						
09:36:20	53104	1866812	97.23	9492	5694032	94.14
09:36:26	1761968	157948	8.23	4224	101548	1.68

ogni amministratore di sistema faccia pratica con le utility di sysstat, in questo modo prenderete confidenza con lo strumento e con il suo output e imparerete a capire come utilizzare i dati al meglio per massimizzare l'efficienza. Utilizzare questi comandi durante un test è un modo semplice per fare pratica e ottenere risultati. Lo spazio disponibile per questo articolo restringe la quantità di dati che possiamo mostrare in questo tutorial, ma per essere sicuri che abbiate tutte le informazioni possibili per favorire la vostra comprensione di queste utility abbiamo incluso l'intero set di dati del nostro scenario all'interno del DVD, inoltre potete trovarli in Rete seguendo i link nelle tabelle.

Il test

Il primo scenario di test consuma risorse della CPU (guardate il file **cpu_loaded.txt** nel DVD oppure il box relativo). Entrambi i campi %user e %system mostrano il tempo di utilizzo della CPU da parte del sistema o da parte degli utenti. La colonna %idle ci dice quando la CPU è scarica. Il campo %iowait è il tempo che la CPU passa aspettando che le richieste I/O siano completate. Un tempo di %idle vicino a zero indica che la CPU sta creando un collo di bottiglia, mentre un alto valore di %iowait indica possibili problemi sulle performance dei dischi. Esaminando i dati aggiuntivi potrete vedere l'impatto sui core mano a mano che lo script progredisce. Questo rapporto, dopo aver eseguito lo script, indica che diverse risorse della CPU sono state utilizzate. Il tempo di %idle è basso.
E guardando il campo %iowait possiamo notare che non contiene dati in questo scenario perché lo script ha come bersaglio la CPU. Il che ci suggerisce che dovremmo studiare un altro po' ma che probabilmente ci troviamo di fronte a un problema di sovraccarico di CPU. Guardando il report di statistica sulla memoria (guardate il file **memory_memeater.txt** sul DVD o il relativo box) ci dobbiamo focalizzare sui campi kbmemfree e kbmemused che sono rispettivamente la parte utilizzata della memoria e quella libera. Potete notare l'enorme diminuzione del contatore di kbmemfree (memoria libera). Qualcosa ha catturato un grande quantitativo di memoria e poi lo ha liberato. Ora abbiamo delle informazioni aggiuntive che ci spiegano anche i cambiamenti sul carico della CPU.
Dal DVD potrete anche esaminare le statistiche di swap nel file chiamato **memory_memeater.txt**; date un'occhiata al campo %swpused. Se questo campo è a zero significa che non è in uso memoria di swap. Un sistema inizia a utilizzare lo swap quando esaurisce la memoria disponibile. Quando utilizza lo swap copia una parte della memoria fisica sul disco fisso; la memoria di swap (il disco) è molto più lenta della memoria fisica e spostare i dati da una memoria all'altra richiede tempo. Un incremento nello swap ha un grande impatto sulle performance. Rivediamo i dati alla luce di quello che abbiamo scoperto. Lo script consuma sempre più memoria libera mentre è in esecuzione. Quando la memoria libera si esaurisce subentra quella di swap. Più memoria viene consumata più swap viene utilizzato. La CPU sta lavorando, la memoria si consuma e lo swap cresce il che ha un impatto negativo sulle performance. Non scoraggiatevi se vi sentite sopraffatti. Risolvere i problemi di performance dei sistemi è molto impegnativo. Studiare i dati per dargli un senso richiede pratica ed esperienza. Lo scenario finale non è generato da uno script ma da un problema reale. I dati sono stati raccolti da un sistema configurato in modo errato. Speriamo che l'analisi dei dati di questo scenario vi aiuti a chiarire un po' meglio le informazioni e a prendere maggior confidenza con le utility di sysstat, nel caso in cui vi siano rimasti dei dubbi. I dati relativi alle 12:46:14 (guardate **nic_sar_data_full_duplex.txt** sul DVD) mostrano i dettagli di rete del kernel durante un trasferimento di un file tra due computer. Invece quelli dell 13:38:59 (guardate **nic_sar_data_half_duplex.txt** sul DVD) mostrano i dettagli di rete del kernel durante la trasmissione di un file tra due computer con problemi di performance. Il campo coll/s (collisioni al secondo) viene popolato in base alle collisioni, cioè quando due (o più) device tentano di inviare dati tramite lo stesso percorso di trasmissione nello stesso momento. Le collisioni producono dati errati che devono venire corretti attraverso un nuovo invio degli stessi dati. Qualcosa sta causando molte collisioni nell'host. Le collisioni rallentano la velocità della rete in modo considerabile; in una rete ideale non dovrebbero mai essercene. Per generare questo problema abbiamo impostato l'interfaccia di rete (NIC) su di un server in modalità 100Mb half duplex, invece del default 100Mb full duplex. La cattiva configurazione della scheda ha un impatto significativo sulla velocità di trasmissione dei dati.

Conclusione

Eccoci qui, ora avete un tutorial sulle utility che esaminano i contatori del kernel. Il kernel di Linux possiede dei contatori interni che tengono traccia delle richieste, dei tempi di computazione, dei blocchi di I/O e di molte altre informazioni interessanti che aspettano solo di essere analizzate e se utilizzate in modo consono, le utility sysstat possono aiutarvi a farlo. Uno dei problemi legati allo studio dei dati sulle performance è che bisogna possedere la conoscenza necessaria per capire come funzionano prima di capire come utilizzarli. Utilizzare le diverse utility per raccogliere informazioni e analizzarle durante un test è un ottimo modo per fare pratica con questi dati prima di avere la necessità di scavare dentro a un problema reale. Ricordate che se cron era già configurato per invocare le utility sysstat potreste avere dei dati pronti per essere studiati. LXP

La risposta

Non esiste una risposta "giusta" alla domanda che abbiamo posto all'inizio del tutorial, ma alcune risposte sono più corrette di altre.
I problemi di performance a volte sono difficili da risolvere. Per formulare una risposta dovete suddividere il sistema in quattro parti. Hardware, software, rete e utenti. L'hardware comprende tutte le componenti del server. Questo include anche eventuali periferiche esterne. Il software è rappresentato dal sistema operativo, il sistema database e le applicazioni. La rete include tutti i componenti di rete compresi tra l'host e il server. Dopo aver fatto una lista delle quattro aree, bisogna descrivere come ognuna di esse impatta sulle performance. I dischi hardware, se lenti, impattano sulle performance di lettura e scrittura. Anche la cattiva o assente indicizzazione delle risorse database impatta sulle performance. Sostituire un hard disk per velocizzare il sistema non porterà grandi risultati se il problema sono gli indici. Potreste notare dei miglioramenti ma probabilmente continuerete a sentire lamentele in certi momenti.

SystemTap: il monitoraggio

Ecco tutto quello che vi serve per cominciare a usare **SystemTap** su Linux



Questo articolo riguarda SystemTap, che fornisce una semplice interfaccia a riga di comando e un linguaggio di scripting per la scrittura di strumentazione per un kernel in esecuzione, così come applicazioni che girano in spazio utente. SystemTap è un software avanzato ed è importante capire che per sondare un sistema "vivente" non dovete ricompilare codice, reinstallare degli eseguibili o riavviare il sistema. In una macchina Ubuntu, potete installare SystemTap eseguendo **apt-get install systemtap systemtap-server systemtap-sdt-dev systemtap-common systemtap-runtime** come root. Potete capire la vostra versione di SystemTap con il comando **stap —version**. Dovreste anche avere i pacchetti **linux-kbuild-<kernel_version>** e **linux-image-<kernel_version>-<architecture>-dbg** (entrambi i nomi dipendono dalla vostra versione del kernel) già installati (Immagine 1). Dovrete anche avere le informazioni di debug incluse nel kernel per permettere a SystemTap di leggere e visualizzare le informazioni. Talvolta ciò significa dover ricompilare il kernel! In un sistema Ubuntu potete aggirare il problema scaricando i file con i simboli adatti al vostro kernel da **http://ddebs.ubuntu.com/pool/main/l/linux** e installandolo (**# dpkg -i <ddeb FILE>**). Se non riuscite a trovare un file di simboli adatto, potete aggiornare il kernel o ricompilarlo. Non dimenticate di riavviare il sistema per applicare le modifiche. Se avete altri errori, potete usare il comando **/usr/bin/stap-prep** per ulteriori indizi. Il comando principale che userete è **stap**. Se provate a lanciare **stap** come utente normale, non vi verrà permesso. Il messaggio d'errore seguente ve ne illustrerà il motivo:

```
You are trying to run systemtap as a normal user.
You should either be root, or be part of the group
"stapusr" and possibly one of the groups "stapsys" or
"stapdev". [man stap]
```

Dal momento che non è considerata buona pratica lavorare come root, è un'idea migliore aggiungere gli utenti a cui volete sia permesso l'uso di stap ai gruppi specificati. Dopo aver sistemato il suddetto errore lanciando **sudo usermod -a -G stapdev,stapusr,stapsys ciromattia**, è il momento di lanciare il processo server **systemtap**. Se provate a lanciarlo come root, SystemTap vi impedirà di farlo lamentandosi: "For security reasons, invocation of stap-serverd as root is not supported". Lanciate quindi il server usando un account utente. Dal momento che SystemTap scrive i dati in due directory, dovrete anche lanciare i seguenti comandi come root (adattando il vostro username) prima che il processo parta con successo:

```
# chown -R ciromattia:ciromattia /var/run/stap-server
# chown -R ciromattia:ciromattia /var/log/stap-server
```

Se le due directory non esistono, createle come root prima di modificarne i permessi. Ora potrete lanciare (**stap-server start**) e fermare (**stap-server stop**) lo **stap-server** senza messaggi d'errore di sorta. Dal momento che accederete a molte risorse diverse, inclusi alcuni dispositivi di sistema, è buona norma lanciare tutti i comandi di SystemTap come root via **sudo**. In ogni caso il proprietario del processo server dovrà essere un utente regolare. Se è già in esecuzione un processo server SystemTap, il seguente output sarà un'indicazione chiara del buon funzionamento nel vostro sistema:

```
$ sudo stap -v -e 'probe kernel.function("sys_open")
{log("Hello World!") exit()}'
```

Tip

Se amministrate solo sistemi Linux, imparare a usare **SystemTap** e **perf_events** è quasi obbligatorio. Se invece gestite sistemi UNIX misti, probabilmente **DTrace** è una scelta migliore.

Come funziona SystemTap

SystemTap ha sonde (chiamate probe) e tapset, che sono gruppi di sonde. Le sonde rilevano eventi e i corrispettivi gestori. SystemTap può osservare a più livelli, dal kernel, librerie e applicazioni fino alle transazioni del database. Alcuni semplici esempi di tapset sono **syscall**, **ioblock**, **memory**, **scsi**, **networking**, **tcp** e **socket**. Per controllare tutte le chiamate di sistema, dovreste scrivere **probe syscall.***. Il comando **stap -L 'syscall.*'** elenca tutti i punti di sonda relativi al set **syscall**. SystemTap controlla anzitutto lo script fornito secondo la libreria del tapset esistente, solitamente posizionata in **/usr/share/systemtap/tapset**, per l'uso di tapset; quindi procede a sostituire tutti i tapset rilevati con le relative definizioni interne alla libreria. SystemTap traduce lo script fornito in C creando un modulo del kernel con l'aiuto del compilatore; in seguito, carica il modulo e abilita tutte le sonde (eventi e gestori) nello script con l'aiuto dell'utility **staprun**. Mano a mano che si verificano eventi nel vostro sistema, vengono eseguiti automaticamente tutti i gestori corrispondenti. Infine, una volta terminata la sessione di SystemTap, tutte le sonde vengono subito disabilitate e il modulo del kernel viene scaricato. Per la lista completa dei tapset, fate riferimento a **https://sourceware.org/systemtap/tapsets**.

```
Pass 1: parsed user script and 95 library script(s) using 90552virt/27192res/2520shr/25500data kb, in 110usr/10sys/118real ms.
...
Pass 5: starting run. Hello World!
Pass 5: run completed in 10usr/20sys/4630real ms.
```

Le righe che cominciano con **Pass 5:** indicano che SystemTap è stato in grado di creare correttamente la strumentazione per sondare il kernel, lanciare la strumentazione, individuare l'evento sondato ed eseguire un gestore valido.

Il linguaggio di scripting di SystemTap

SystemTap supporta un linguaggio di scripting simile a AWK. Ci sono due componenti principali: sonde (probe) e funzioni. In queste le istruzioni e espressioni usano una sintassi di operatori e precedenza simile a C. Tornando all'esempio "Hello World!" eseguito poc'anzi, l'evento **kernel.function("sys_open")** scatena il gestore racchiuso tra **{}**. Il gestore stampa semplicemente il messaggio "Hello World!" ed esce. In luogo di **log()** potevate usare la più versatile **printf()**: quest'ultima in SystemTap è molto simile alla controparte in C. Un'altra funzione utile è **target()**: viene usata per indicare a uno script di prendere come argomento un ID di processo o un comando. Ha lo stesso effetto di specificare **if (pid() == IDprocesso)** ogni volta che volete analizzare un processo specifico. L'uso di **target()** rende i vostri script più versatili, flessibili e riusabili dal momento che potete passare l'ID di un processo come argomento ogni volta che volete lanciare gli script stessi.

Sonde multiple

Uno script SystemTap può avere più sonde e potete lanciare i comandi in due modi. Il primo metodo è inserendo il codice SystemTap in riga di comando e il secondo è utilizzare un file per memorizzare il codice ed eseguirlo da qui. Il primo è più veloce ma il secondo è migliore, specialmente se volete eseguire comandi multipli. Il comando **stap** esegue il programma **staprun** che è il backend di SystemTap. **staprun** si aspetta un modulo kernel prodotto dal comando **stap**. Il codice seguente illustra le varie parti di un programma **stap**:

```
$ cat example.stp
probe begin { printf("Inizio!\n") exit() }
probe * { printf("Qui succedono cose!\n") exit() }
probe end { printf("Fine!\n") exit() }
$ sudo stap example.stp
Inizio!
Qui succedono cose!
Fine!
```

Come potrete immaginare, la parte **begin** viene sempre eseguita per prima e la parte **end** per ultima. La parte **probe *** indica a SystemTap di reagire a ogni controllo. Se **probe *** non ha una condizione d'uscita, dovrete terminarlo manualmente premendo **Control**+**C**.

Disegnare grafici

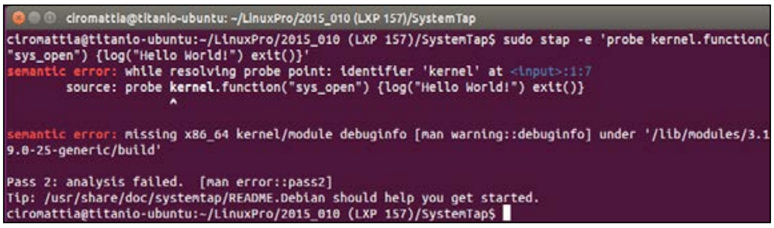
Gli istogrammi sono molto pratici per ottenere una panoramica di alto livello su quello che succede nel vostro sistema. SystemTap può generare istogrammi con l'aiuto della funzione **@hist_log()**, eseguita dopo la collezione dei dati. Il comando seguente colleziona dati per la chiamata di sistema **write()** e genera un istogramma in potenza di due per le dimensioni della write ritornata:

```
$ sudo stap -ve 'global data; probe syscall.write.return { data <<< $return; } probe end { printf("\n\trval (bytes)\n"); print(@hist_log(data)); }'
...
            rval (bytes)
   value |-------------------------------------------------- count
       0 |                                                    0
       1 |@@@@@@@@@@@                                        46
       2 |@@@@                                               16
...
```

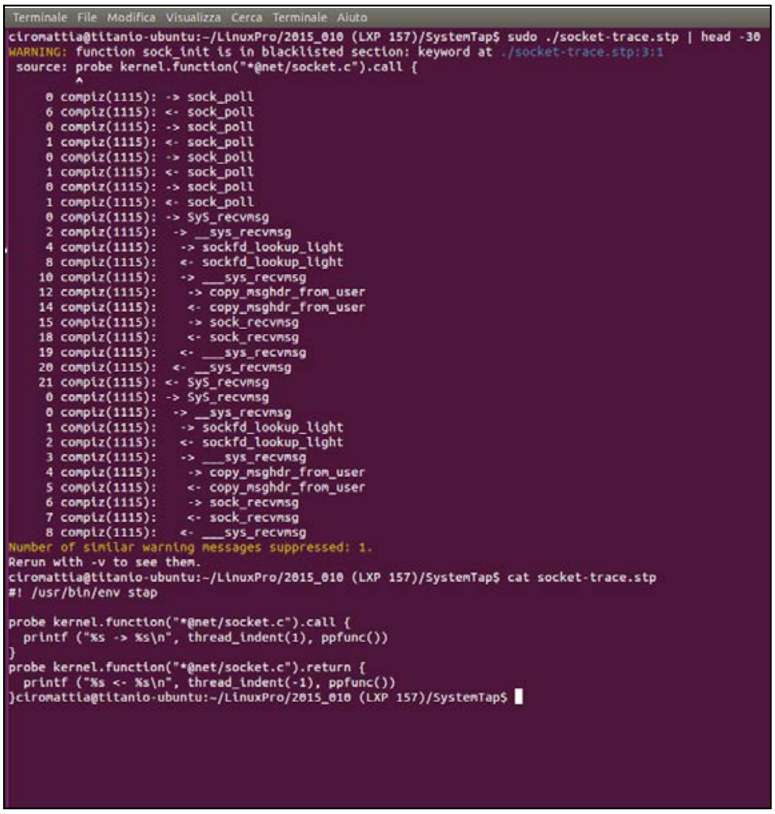
Una variabile globale chiamata **data** viene usata per »

Tip

Potete scoprire di più su SystemTap all'indirizzo **https://sourceware.org/systemtap**. Un ottimo libro sulle performance di sistema che affronta anche SystemTap è **Systems Performance: Enterprise and the Cloud** di Brendan Gregg. Potreste anche leggere la documentazione su **https://access.redhat.com/documentation/en-US/Red_Hat-Enterprise_Linux**.



› Immagine 1. Se ci sono pacchetti mancanti, *SystemTap* non funzionerà correttamente. Usate i messaggi d'errore per capire la radice del problema e correggerlo



› **Lanciando lo script *socket-trace.stp* in un sistema Linux genererà un output simile a quello presentato in quest'immagine. Socket-trace. stp è pratico per capire come ogni processo interagisce con la rete a livello kernel**

» archiviare i dati acquisiti. Dopo la collezione di dati, la parte **end** si occupa di generare e stampare l'istogramma.

Grafici a fiamma

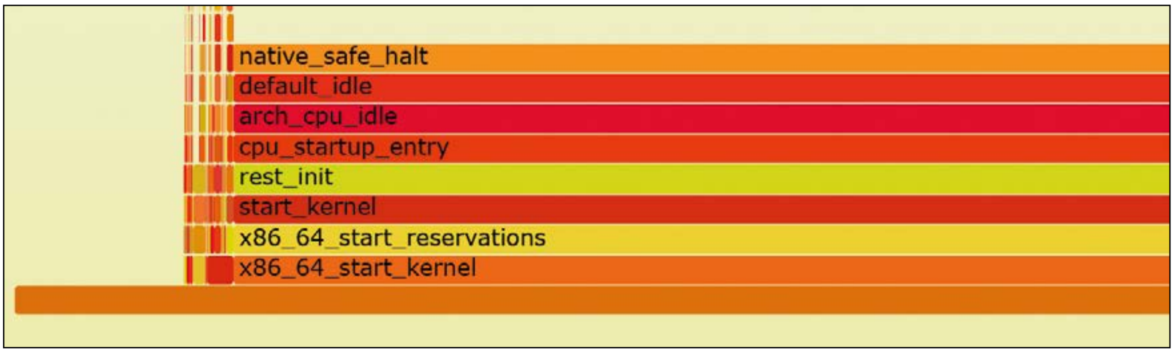
I grafici a fiamma, creati da Brendan Gregg, sono un modo eccellente per presentare dati catturati con SystemTap. Visualizzano il software profilato in modo da permettere la facile identificazione della maggior parte dei percorsi di codice. Usando un grafico a fiamme, un amministratore può tracciare visualmente quali funzioni sono chiamate da altre funzioni. Righe diverse rivelano quali funzioni di un programma individuale stanno consumando una quantità sproporzionata di tempo CPU. Tali visualizzazioni vi mostrano facilmente il grande disegno. Vi serviranno due programmi esterni per generare un grafico a fiamme, ma l'intero processo è relativamente facile da seguire. I comandi seguenti illustrano il processo:

```
$ sudo stap -s 32 -D MAXBACKTRACE=100 -D
MAXSTRINGLEN=4096 -D MAXMAPENTRIES=10240
-D MAXACTION=10000 -D STP_OVERLOAD_
THRESHOLD=5000000000 --all-modules -ve 'global s;
probe timer.profile { s[backtrace()] <<< 1; } probe end {
foreach (i in s+) { print_stack(i); printf("\t%d\n", @
count(s[i])); } } probe timer.s(60) { exit(); }'
> LXP.stap
...
$ head -5 LXP.stap
0xffffffff811e1c54 : mangle_path+0xc4/0xd0 [kernel]
...
0xffffffff81221dd3 : show_pid_map+0x13/0x20 [kernel]
$ ./stackcollapse-stap.pl LXP.stap > LXP.folded
$ cat LXP.folded | ./flamegraph.pl > LXP.svg
```

Le opzioni importanti di SystemTap necessarie per la corretta generazione del grafico sono **-D MAXBACKTRACE=100** e **-D MAXSTRINGLEN=4096** dal momento che vi assicurano che gli stack trace non vengano troncati. Notate che il comando **stap** precedente uscirà automaticamente, quindi non dovrete fermarlo con **Ctrl+C**. Recuperate i dati voluti, processateli e generate il grafico a fiamma come file SVG. Potete recuperare entrambi gli script da **https://github.com/brendangregg/FlameGraph**. L'immagine sotto mostra l'immagine del grafico generato. Generare il grafico a fiamma è facile; la parte difficile è la corretta interpretazione dei risultati, che dipende dalla vostra esperienza e dalla vostra buona conoscenza del sistema Linux che state controllando. Potete usare i grafici a fiamma anche per visualizzare dati creati da altri strumenti, inclusi **DTrace**, **Windows XPerf**, **perf_events**, **OS X Instruments** e **Google Chrome Developer Tools**.

Script socket

C'è di buono che potete trovare script utili e pronti nel sito di SystemTap. Per esempio, lo script **socket-trace.stp** traccia le funzioni chiamate dal file del kernel **net/socket.c**. L'immagine sulla sinistra mostra una piccola parte dell'output prodotto dallo script su Ubuntu. L'output è a livello piuttosto basso; sarebbe stato difficile ottenere in altro modo tali informazioni, specialmente in applicazioni per le quali non avete accesso al sorgente. Lo script **iotime.stp** traccia ogni volta che una chiamata di sistema apre, chiude, legge o scrive su un file. Potete fermarlo manualmente con



› **Un grafico a fiamme è un ottimo modo per visualizzare i dati raccolti usando SystemTap**

Opzioni da riga di comando utili

SystemTap supporta molte opzioni da riga di comando. Eccone alcune tra le più utili; per una lista completa dovreste controllare la pagina di manuale per **stap**. L'opzione **-s** seguita da un numero indica a SystemTap di usare 'numero' megabyte di buffer per i trasferimenti di dati kernel-to-user. Se avete un multiprocessore in modalità Bulk, è un totale per-processore. L'opzione **-d** seguito da una coppia nome-valore indica di aggiungere la direttiva del preprocessore C indicata al **Makefile** del modulo. Viene spesso usata per reimpostare i parametri limite. L'opzione **-d** seguita dal nome di un modulo indica a SystemTap di aggiungere un simbolo o informazioni per il modulo kernel specificato. L'opzione **—all-modules** è equivalente allo specificare **-dkernel** e **-d** per ogni modulo kernel attualmente caricato. Quest'opzione può ingrossare sensibilmente i moduli sonda. L'opzione **-e** indica a SystemTap di aspettare e lanciare uno script passato a riga di comando. Se volete far leggere a **stap** uno script da un file, dovete specificarne il nome a riga di comando senza ulteriori parametri. L'opzione **-o** seguita da un nome di file indica di inviare lo standard output a un file chiamato **nomefile**. L'opzione **-v** specifica a SystemTap di incrementare la verbosità delle informazioni prodotte per ogni passaggio. L'opzione **-c** seguita da un comando specificato con il percorso completo imposta la funzione di gestione **target()** al comando specificato. Allo stesso modo, l'opzione **-x** seguita dall'ID di un processo indica a SystemTap di impostare la funzione di gestione **target()** all'ID di processo specificato. Ora dovrebbe essere facile capire che il comando **stap** seguente è analogo al comando **strace**:

```
$ sudo stap -c /bin/ls -ve 'probe syscall.* {printf("PID: %d\tNAME: %s\tARGSTR: %s\n",pid(), name, argstr); }'
```

Control+**C**. La cosa veramente utile di questo script è che scrive il nome del processo che ha provocato l'azione sul file, il suo ID di processo e il nome del file aperto. Se un processo è riuscito a scrivere o leggere da un file, una coppia di righe **access** e **iotime** appariranno assieme nell'output. I dati presentati sono a un livello piuttosto basso e potete filtrare e modificare l'output usando il comando **grep**. L'immagine 2 mostra una piccola sezione dell'output dello script **iotime.stp**. Potete anche controllare file individuali con l'aiuto dello script **inodewatch.stp**. Dovrete scoprire alcune informazioni a più basso livello, prima, con l'aiuto del comando **stat**. L'output seguente illustra il processo di monitoraggio su **/var/log/wtmp**:

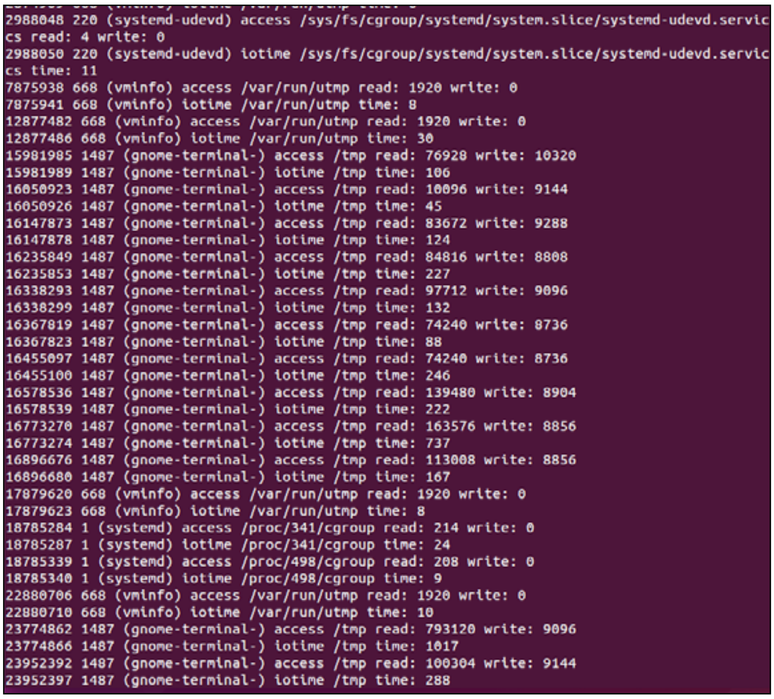
```
$ stat -c '%D %i' /var/log/wtmp
801 813532
$ sudo ./inodewatch.stp 0x8 0x01 813532
accounts-daemon(1574) vfs_read 0x800001/813532
last(11805) vfs_read 0x800001/813532
```

Il comando **stat** ritorna le informazioni richieste per lanciare **inodewatch.stp**: **813532** è il numero di inode che volete controllare, mentre **8** è il numero major del dispositivo e **01** è il numero minor del dispositivo nel filesystem che contiene il file stesso. Il prefisso **0x** davanti ai numeri indica valori in formato esadecimale.

Script per processi

Lo script **topsys.stp** elenca le prime 20 chiamate usate dal sistema in ogni intervallo di cinque secondi, così come il totale delle volte in cui la singola chiamata è stata usata in tale periodo. Potete modificare l'intervallo cambiando la riga **probe timer.s(5)"**. Modificate il numero di chiamate di sistema stampate cambiando la riga **foreach (syscall in syscalls_count - limit 20)**. Lo script **syscalls_by_proc.stp** elenca i primi 20 processi che stanno facendo il maggior numero di chiamate di sistema. Mostra anche quante chiamate ogni processo ha fatto durante l'intervallo di tempo. L'output dovrebbe essere simile a questo:

```
#SysCalls    Nome processo
20352        Xorg
3015         compiz
768          mysql
```

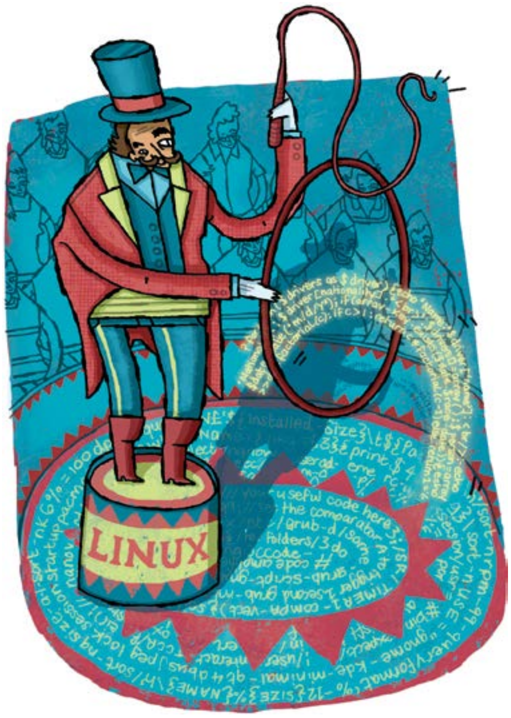


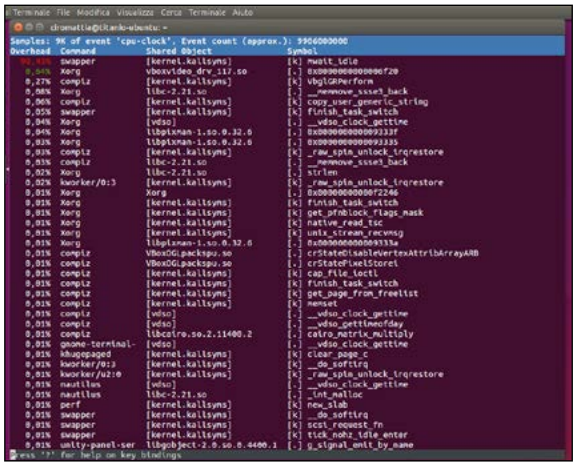
› Questa immagine mostra lo script iotime.stp su un sistema Linux. iotime.stp è utile per tracciare quando una chiamata di sistema apre, chiude, legge o scrive un file

SystemTap può anche operare a livello di rete. Lo script **net.stp** mostra informazioni riguardo la rete per interfaccia e per processo. Altri script interessanti da esaminare sono **iostats.stp**, **timeout.stp** e **stopwatches.stp**. Tutti gli script SystemTap menzionati sono utili per ogni amministratore di sistema e di rete, a seconda del problema che deve gestire. Il modo più semplice di sviluppare i vostri script è cominciare da uno esistente e modificarlo. Se installate il pacchetto **systemtap-doc** vi verranno installati la maggior parte degli script in **/usr/share/doc/systemtap-doc/examples**. Il problema principale di strumenti come SystemTap è che dovete usarli regolarmente per diventarne padroni, dal momento che "il momento giusto per riparare il tetto è quando splende il sole". In altre parole, non provate a imparare SystemTap quando dovete risolvere un problema vero! LXP

Come tracciare il kernel

Come iniziare a usare **perf_events** per analizzare le performance del vostro sistema Linux





› **Questa è la prima schermata dell'output del comando *perf report* con i dati raccolti da perf record -a -F 1000 sleep 10. Potete ottenere più informazioni riguardo ogni riga evidenziandola e premendo Invio**

Questo mese affronterete perf_events, uno strumento per la profilazione e tracciamento in Linux. La ragione più importante per la profilazione in Linux è comprendere l'utilizzo della CPU in modo rapido e veloce. L'affinamento delle performance non vi permette solo di risolvere problemi, ma vi renderà in grado di sfruttare il vostro hardware più efficientemente senza spreco di risorse. Il comando **perf**, originariamente chiamato *Performance Counters for Linux* (PCL), è una raccolta di strumenti per la profilazione e il tracciamento che oggi è chiamato **Linux Performance Events** (LPE). Ogni strumento separato è utilizzato come sottocomando di **perf**, ma non li vedrete tutti, privilegiando invece i più utili. Talvolta la parte più difficile nell'usarli è decidere quale sia il sottocomando giusto per quello che dovete fare, e se doveste aver bisogno di informazioni su un preciso sottocomando usate `perf help <subcommand_name>`. Installare **perf_events** è facile dal momento che è parte del kernel, quindi su un sistema Ubuntu con Linux 3.19.0-25 potete installarlo con

```
$ sudo apt-get install linux-tools-common linux-tools-3.19.0-25-generic linux-cloud-tools-3.19.0-25-generic
```

il che installerà tutti i pacchetti richiesti. Nota: come indovinerete, il nome esatto dei pacchetti da installare va in base alla versione del kernel. Perf_events non è solo più facile da installare rispetto a **DTrace** e **SystemTap**, ma anche più facile da imparare. Tuttavia tutti e tre gli strumenti necessitano di un po' di pratica prima di essere usati efficientemente. Dopo l'installazione potete usare perf_events con il comando **perf**. Praticamente tutti i sottocomandi necessitano di privilegi di root per girare, ma per conoscerne la versione potete usare `perf --version` senza privilegi di root. Perf_events passa dati al livello utente per il post-processing, usando un framework di scripting che potrebbe aggiungere un po' di overhead al processo, specialmente nel tracciare eventi frequenti, mentre sia DTrace sia SystemTap possono catturare dati e processarli allo stesso tempo. Perf_events supporta il tracciamento sia statico sia dinamico. Il primo può essere fatto usando un qualsiasi punto di tracciamento disponibile, mentre quello dinamico viene eseguito definendo e creando delle sonde personalizzate dal codice kernel al volo! L'output seguente mostra il numero di punti

Tip

Se amministrate solo sistemi Linux, imparare **SystemTap** e **perf_events** è quasi obbligatorio. Se invece gestite sistemi UNIX misti, probabilmente **DTrace** è una scelta migliore.

Fuoco e fiamme (e grafici)

Vale la pena imparare qualcosa riguardo i grafici a fiamma dal momento che hanno alcune caratteristiche interessanti e uniche. Questi grafici possono visualizzare frame di stack profilati e permettervi di capire come viene sfruttata una CPU in modo veloce e utile. Ogni casella in un grafico a fiamma rappresenta una funzione dello stack (in figura nella pagina successiva). I valori nell'asse y sono il numero di frame nello stack. Le funzioni nel margine superiore sono funzioni eseguite direttamente dalla CPU. La funzione sotto a un'altra è il padre, rappresentato come stack. Il valore nell'asse x rappresenta la popolazione campionata. Una casella più ampia indica tre cose: la funzione è stata chiamata più spesso o è risultata più lenta delle caselle più piccole, o una combinazione di queste due possibilità. Se diversi programmi sono in esecuzione durante il campionamento, la base del grafico avrà più caselle distinte. Un grafico a fiamma non mostra il totale delle chiamate o il passare del tempo, dal momento che l'ordinamento sinistra-destra non ha significato a parte l'ordine alfabetico. In più, i colori nel grafico non hanno significato particolare se non quello di essere esteticamente gradevoli. I grafici SVG generati hanno un'altra caratteristica interessante: sono interattivi; se passate il mouse sopra i vari elementi, il JavaScript integrato rivelerà dettagli in fondo all'immagine. I grafici a fiamma, come altre grafiche, risolvono problemi reali aiutandovi a capire e interpretarne i risultati. Vi risparmiano anche di controllare gli affollati dati dello stack trace. Possono anche essere usati per visualizzare memoria, I/O, ecc.

di eventi di tracciamento per set:

```
$ sudo perf list | grep "Tracepoint event" | awk -F: {'print $1'} |
sort | uniq -c | sort -rn
    596   syscalls
    141   cfg80211
     93   ext4
     35   xen
     25   writeback
…
```

Come potete facilmente capire, la maggioranza dei tracepoint sono relativi alle chiamate di sistema.

Usare perf_events

Perf_events opera in due fasi: la prima fase riguarda la raccolta di dati e genera un file con questi ultimi mentre il secondo riguarda l'analisi dei dati e il processing del file stesso.

```
$ sudo perf record -a -F 1000 sleep 10
[ perf record: Woken up 2 times to write data ]
[ perf record: Captured and wrote 1.051 MB perf.data (~45904
samples) ]
ciromattia@titanio-ubuntu:~$ ls -l
totale 1132
-rw------- 1 root      root      1104468 ago  4 18:29 perf.data
```

Dovrete lanciare il comando come root con l'aiuto di `sudo`, dal momento che **perf** usa varie risorse di sistema per raccogliere i dati. Il parametro `-a` indica a **perf** di controllare la CPU, mentre `-F` definisce la frequenza di profilazione. Il comando eseguito è `sleep 10`, che indica che i dati saranno catturati per 10 secondi. Se voleste interrompere manualmente l'esecuzione in un qualsiasi momento, potete premere **Control**+**C**. Se avete già un file **perf.data**, quello vecchio sarà rinominato **perf.data.old**. Se è già presente anche quello e lanciate `$ perf record`, il file **perf.data.old** meno recente sarà perso perché sovrascritto dal perf.data rinominato. Potete analizzare un file perf.data usando `$ perf report` (vedete l'output in figura). Le righe di output che includono il testo <not supported> indicano che il kernel in esecuzione manca del supporto per la funzionalità richiesta da **perf** per profilare tali aspetti del vostro programma. Solitamente succede in ambienti virtualizzati.

Recuperare informazioni su I/O

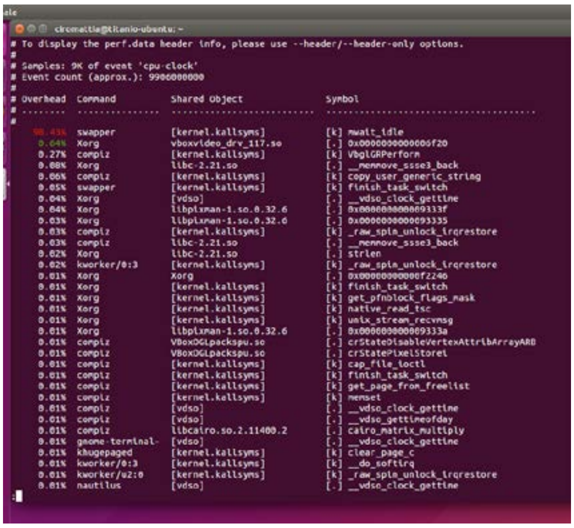
Talvolta i dischi possono avere problemi, e richiedono un controllo sulle performance. La sonda **block:block_rq_issue** viene scatenata ogni volta che viene sollevata una richiesta di I/O su un dispositivo a blocchi. Il comando **perf** seguente traccia la sonda **block:block_rq_issue**:

```
$ sudo perf record -e block:block_rq_issue -ag
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.562 MB perf.data
(~24543 samples) ]
```

Dovreste premere **Ctrl**+**C** per fermare l'esecuzione del summenzionato comando **perf**. Il parametro `-g` indica a perf_events di catturare anche gli stack trace (vedete la schermata qui sotto per un esempio di output di `sudo perf report --stdio -I`). Quello che è realmente importante è che potete vedere sia il comando **perf** utilizzato che le informazioni sull'host, la data, la memoria, la versione del kernel e la CPU della macchina Linux usata durante il processo di cattura. Potete recuperare facilmente informazioni sui page fault come segue:

```
$ sudo perf record -e page-faults -a
$ sudo perf report --stdio
# Samples: 12  of event 'page-faults'
# Event count (approx.): 381
...
    39.63%  Xorg            libpixman-1.so.0.32.6  [.]
0x0000000000072af3
    22.31%  gnome-terminal-  libc-2.21.so          [.] __GI___
mempcpy
    16.54%  Xorg            libpixman-1.so.0.32.6  [.]
```

»



› **Questo output è stato generato con sudo perf report --stdio -I. Il parametro --stdio disabilita la modalità interattiva del sottocomando di report e vi permette di processare o filtrare l'output testuale usando altri comandi Unix**

Tip

Potete sapere di più su perf_events sul wiki ufficiale **http://bit.ly/PerfWiki**. Più imparate riguardo i processi disponibili, più facile diventa, quindi un **sudo perf list** di tanto in tanto vi gioverà.

```
0x0000000000072af7
```

Il primo comando registra tutti i page fault e (ancora) può essere fermato premendo **Ctrl+C**. Lanciare il comando report come $ sudo perf report -n --stdio aggiungerà una colonna con il conteggio dei campioni all'output. L'output mostra quale comando ha causato la maggior parte dei page fault. Doveste voler recuperare tutte le sonde (o tracepoint) in relazione ai page fault, potete eseguire $ sudo perf list | grep faults.

Grafici a fiamma

I grafici a fiamma, creati anch'essi dallo sviluppatore di perf B_ rendan Gregg, sono un modo eccellente di presentare i dati catturati con perf_events. Vi serviranno due script Perl esterni per generare i grafici, ma l'intero processo è relativamente facile da seguire, e l'output è gradevole ed esaustivo. I comandi **perf** seguenti illustrano il processo:

```
$ sudo perf record -F 99 -a -g -- sleep 60
$ sudo perf script | ./stackcollapse-perf.pl > flame.perf
$ ./flamegraph.pl flame.perf > perfFlame.svg
```

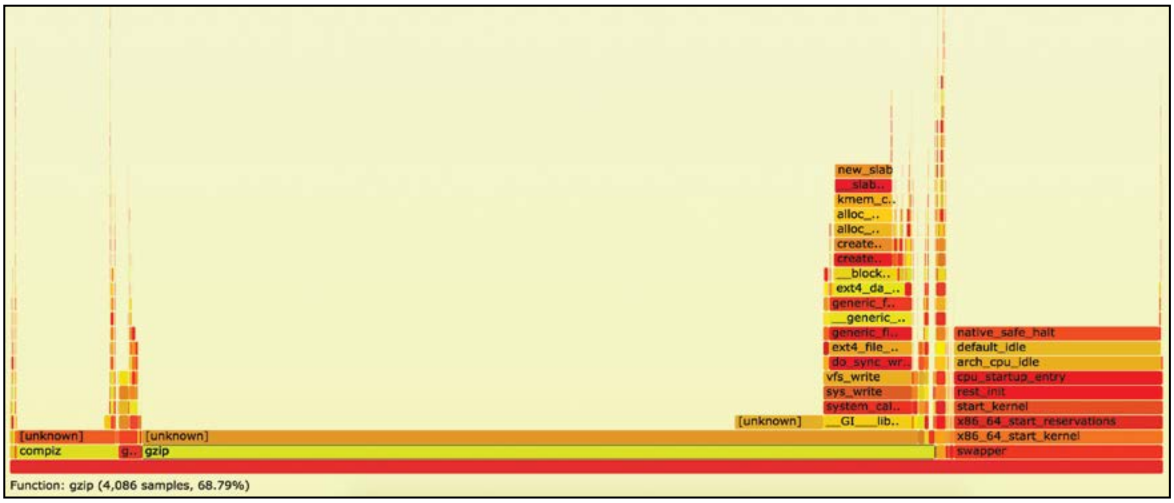
L'output generato varierà a seconda delle informazioni recuperate (vedete sotto per output di esempio). Spiegare un grafico a fiamma non è sempre un compito semplice ma migliora con l'esperienza. Potete recuperare entrambi gli script (**stackcollapse-perf.pl** e **flamegraph.pl**) da **https://github.com/brendangregg/FlameGraph**. Tenete a mente che se il vostro kernel Linux non contiene le informazioni di debug necessarie, potreste ottenere i seguenti messaggi d'avviso nell'eseguire $ sudo perf script. Ciò indica che il grafico conterrà informazioni meno dettagliate:

```
no symbols found in /bin/gzip, maybe install a debug package?
Failed to open /tmp/perf-2294.map, continuing without... no symbols found in /usr/lib/x86_64-linux-gnu/dri/swrast_ dri.so, maybe install a debug package?
no symbols found in /usr/sbin/VBoxService, maybe install a debug package?
```

Creare un istogramma

Gli istogrammi sono un altro modo molto pratico di ottenere una panoramica ad alto livello di ciò che succede nel vostro sistema. Anche se perf_events non supporta direttamente gli istogrammi e l'aggregazione, possono essere facilmente generati con l'aiuto di uno script shell scritto dal solito Brendan Gregg. Sarebbe bello dire che generare un istogramma con perf_events è facile e personalizzabile quanto farlo con DTrace o SystemTap, ma così non è. L'output seguente mostra il processo, quando ci provate:

```
$ sudo ./perf-stat-hist syscalls:sys_enter_write count 15
Tracing syscalls:sys_enter_write, power-of-4, max 1048576,
for 15 seconds...
```



› **Un grafico a fiamma è un modo eccellente per visualizzare dati acquisiti da *perf_events*. Potete anche usarlo per dati creati con altri strumenti, inclusi DTrace, OS X Instruments, SystemTap e Google Chorme Developer Tools**

Di più riguardo i vari sottocomandi

Se lanciate il comando **perf** senza altri argomenti o opzioni, vedrete un messaggio di aiuto che include tutti i sottocomandi disponibili, eccoli:

» **kmem** ha due varianti: eseguito come perf kmem stat riporta statistiche sulla memoria a livello di sistema, mentre eseguito come perf perf kmem record <comando> registra gli eventi kmem di un comando specifico.

» **diff** legge due file **perf.data** e ne mostra le differenze.

» **stat** esegue un comando e ne raccoglie le statistiche di performance. Questo sottocomando non genera un file **perf.data**, ma ritorna immediatamente le informazioni richieste.

» **trace** mostra gli eventi associati al target, incluse le syscall, e altri eventi di sistema come i page fault. Questo sottocomando non genera automaticamente un file **perf.data** e può essere usato in sostituzione per il comando **strace(1)**.

» **record** serve a controllare l'attività di CPU e genera un file **perf.data** da processare in seguito. Questo sottocomando può essere usato in combinazione con altri sottocomandi.

» **report** è usato per analizzare un file **perf.data** generato in precedenza. Potete usare il parametro -i seguito da un nome di file per specificare il file in input. Se non viene usato il parametro -i, **perf** cercherà **perf.data** all'interno della directory attuale.

» **timechart** è uno strumento per visualizzare il comportamento generale del sistema durante un carico di lavoro. Il file immagine generato è creato in formato SVG.

» **top** è un sottocomando che genera e mostra un contatore di performance profilato in tempo reale. L'output vi ricorderà molto dell'utility **top**.

» **annotate** è un sottocomando che legge un file **perf.data** e ne mostra una versione annotata. Se il file oggetto contiene simboli di debug, il codice sorgente sarà visualizzato a fianco del codice assembly. Se non sono presenti informazioni di debug nell'oggetto, verrà visualizzato solamente l'assembly annotato.

```
         Range     : Count    Distribution
           -> -1   : 0       |                              |
          0 -> 0   : 0       |                              |
          1 -> 3   : 1       |###################           |
          4 -> 15  : 2       |##############################
##########|
 ...
```

Lo script **perf-stat-hist** che vi serve è recuperabile da **https://github.com/brendangregg/perf-tools/blob/master/misc/perf-stat-hist**. La modalità di perf_events necessaria per gli istogrammi è quella 'di conteggio' che riassume gli eventi occorsi nel kernel Linux e passa tale riassunto in spazio utente, per esempio il comando:

```
$ sudo perf stat -e 'syscalls:sys_enter_*' -p 1339
```

conta le chiamate di sistema per il processo con PID 1339, mentre

```
$ sudo perf stat -e 'syscalls:sys_enter_*' -a sleep 15
```

conta le chiamate di sistema per l'intero sistema in un intervallo di 15 secondi. Il pratico

```
$ sudo perf top -e raw_syscalls:sys_enter -ns comm
```

mostra le chiamate di sistema per processo e l'output viene aggiornato ogni due secondi. Il comando seguente conta le chiamate di sistema necessarie per l'esecuzione del comando specificato e stampa i soli valori non-zero con l'aiuto di **awk**:

```
$ sudo perf stat -e 'syscalls:sys_enter_*' gzip <a_
command> 2>&1 | awk '$1 != 0'
```

È meglio usare il sottocomando $ perf stat quando possibile, visto il suo ridotto overhead.

Generare una mappa di calore

Una mappa di calore è un modo intelligente di visualizzare una tabella di numeri, sostituendo i valori numerici con celle colorate. La buona notizia è che potete generare mappe di calore usando i dati recuperati da perf_events con l'aiuto di un altro script Perl. Il processo richiede i passi seguenti:

```
$ sudo perf record -e block:block_rq_issue -e block:block_
rq_ complete -a sleep 120
$ sudo perf script | awk '{ gsub(/:/, "") } $5 ~ /issue/ {
ts[$6, $10] = $4 } $5 ~ /complete/ { if (l = ts[$6, $9]) { printf
"%.f %.f\n", $4 * 1000000, ($4 - l) * 1000000; ts[$6, $10] = 0
} }' > out.perf
$ ./trace2heatmap.pl --unitstime=us --unitslat=us
--maxlat=50000 out.perf > perfHM.svg
```

L'esempio generato (vedete sotto) mostra il progredire del tempo nell'asse x, la latenza nell'asse y mentre l'asse z rappresenta il numero di I/O in quel momento e spettro di latenza, rappresentato da vari colori. Potete recuperare lo script **trace2heatmap.pl** da **https://github.com/brendangregg/HeatMap**. Lanciate $ ./trace2heatmap.pl --help per vedere tutte le opzioni a disposizione.

Creare un nuovo tracepoint

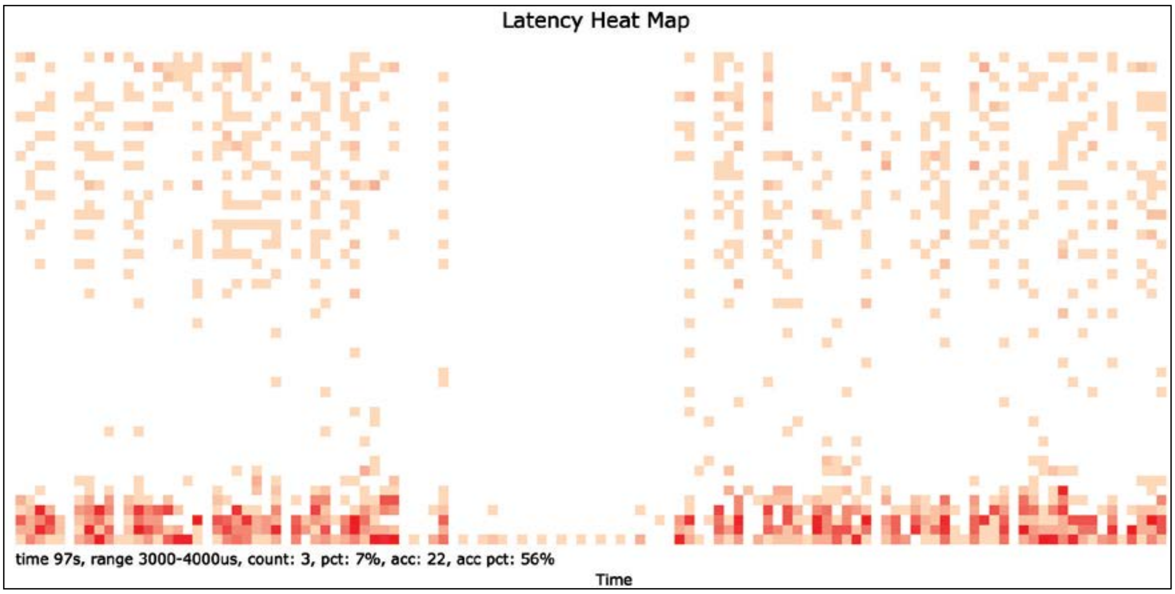
Per concludere, ecco brevemente come potete definire un nuovo tracepoint. Per prima cosa userete il comando $ perf probe --add <nuovo_tracepoint> per aggiungere un nuovo punto di tracciamento chiamato <nuovo_tracepoint> mentre $ perf probe -d <tracepoint_esistente> rimuove un tracepoint esistente. Se volete elencare tutti i tracepoint dinamici, potete usare sudo perf probe -l .
Se il tracepoint che state cercando di aggiungere non esiste nel kernel, otterrete un messaggio d'errore simile al seguente:

```
$ sudo perf probe —add un_tracepoint
Failed to find path of kernel module.
Kernel symbol 'un_tracepoint' not found.
Error: Failed to add events. (-2)
```

Talvolta, per supportare un nuovo tracepoint, avrete bisogno di ricompilare il kernel dai sorgenti per includere le informazioni richieste. Perf_events continua a evolvere e migliorare con il susseguirsi delle versioni. Sarebbe fantastico se avesse tutte le caratteristiche di DTrace ma al momento quest'ultimo è in vantaggio. Tuttavia sta a voi decidere se potete far quello che vi serve con perf_events o se vi servono le potenzialità avanzate e mature di DTrace. Non dimenticate che DTrace è difficile da installare e non pienamente supportato da ogni distribuzione Linux. Configurare le performance e l'identificazione di problemi è simile debug, che è un compito difficile che richiede molta esperienza, quindi siate pazienti e persistenti. Conoscere perf_event è un buon inizio. LXP

Tip

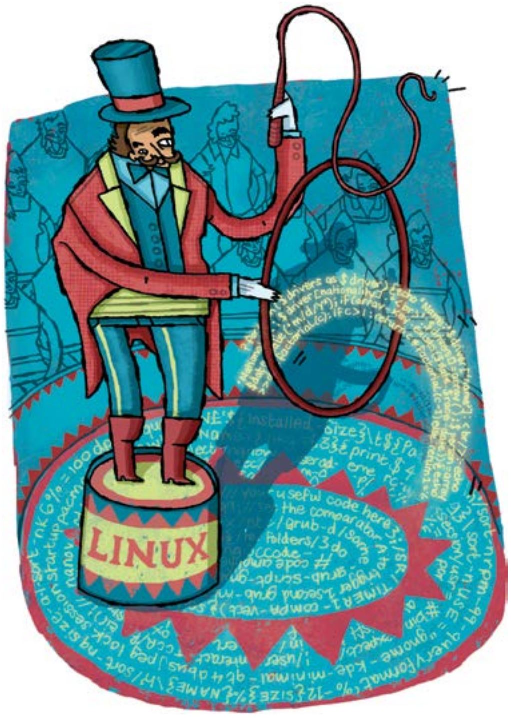
Un ottimo libro sull'analisi delle performance di sistema che copre anche perf_events è **System Performance: Enterprise and the Cloud** di Brendan Gregg.



› Questa è una mappa di calore, un modo alternativo di presentare numeri sullo schermo. I dati recuperati da *perf_events* possono essere trasformati in tale mappa se processati con un programma o script esterno

Abbandonarsi alla funzionalità

Eccovi i concetti basilari che dovete capire per cominciare a programmare con il linguaggio **Scala**



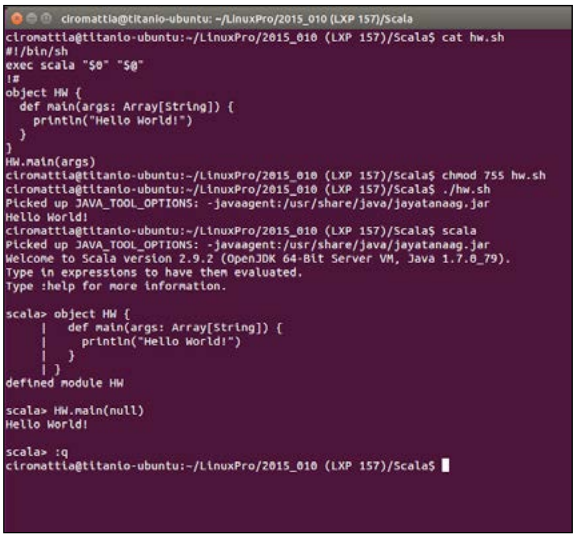
Scala è un linguaggio di programmazione moderno progettato da Martin Odersky. Il nome Scala deriva dalle parole "scalabile" e "linguaggio", e significa che il linguaggio è pensato per crescere con le necessità degli utenti. Questo articolo vi presenterà tecniche utili e caratteristiche di Scala, ed è consigliabile cominciare a sviluppare piccoli programmi in Scala il prima possibile. Nota: se state seguendo questa serie di articoli sui linguaggi di programmazione, potreste aver notato che solitamente viene presentato codice che calcola i numeri di Fibonacci, implementazioni di algoritmi di ordinamento e file I/O. Questo perché sarete così in grado di confrontare come programmi simili vengano implementati in linguaggi diversi.

Installare Scala

Potete installare Scala in un sistema Debian con **apt-get install scala** come root, e ottenerne la versione con **scala -version** dopo l'installazione. Scala fornisce lo Scala code runner (**scala**), che è un ambiente interattivo, e il compilatore

Tip

Potete trovare più informazioni su Scala su **www.scala-lang.org**. Due ottimi libri sono **Programming in Scala** di Martin Odersky e **Scala in Action** di Nilanjan Raychaudhuri.



› Fig 1: Potete lanciare il codice Scala usando lo Scala code runner, oppure integrarlo in uno script shell!

Scala (**scalac**) che vi aiuta a compilare i vostri programmi. Tipicamente userete l'ambiente interattivo per provare cose nuove e sperimentare, e il compilatore per generare file eseguibili. Potete uscire dallo Scala code runner scrivendo **:q** e premendo Invio. Il codice Scala per il classico "Hello world" è:

```
object HW {
        def main(args: Array[String]) {
                println("Hello World!")
        }
}
```

Per prima cosa dovete definire un nuovo oggetto, chiamato HW (potete usare il nome che volete) all'interno del quale implementare la funzione **main()**, dove metterete il codice che stampa il messaggio desiderato. Potete eseguire il vostro codice Scala in tre modi diversi: come uno script, usando lo Scala code runner o con il compilatore. La **Fig 1** mostra il primo e il secondo modo, il terzo funziona come segue:

```
$ scalac hw.scala
$ ls -l
total 12
-rw-r--r-- 1 mtsouk mtsouk 586 Mar 16 09:41 HW.class
-rw-r--r-- 1 mtsouk mtsouk 581 Mar 16 09:41 HW$.class
-rw-r--r-- 1 mtsouk mtsouk 87 Mar 14 21:04 hw.scala
```

```
$ scala HW
Hello World!
```

In più, potete lanciarlo dalla riga di comando UNIX usando lo Scala code runner con **scala hw.scala**. Il vantaggio di tale metodo a confronto dell'uso del compilatore è che non genera file intermedi, ma è più lento.

Calcolare numeri di Fibonacci

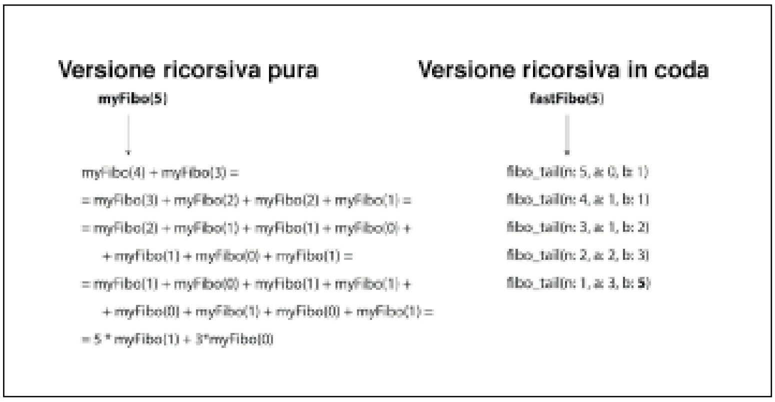
Una prima implementazione per calcolare i numeri di Fibonacci è la seguente:

```
def myFibo( n : Int) : Int = n match {
        case 0 | 1 => n
        case _ => myFibo( n-1 ) + myFibo( n-2 )
}
```

La riga **case _** si applica a tutti i casi non coperti dal primo case del blocco match. È importante che tutti i casi siano coperti, altrimenti il vostro programma potrebbe saltare. Potete far riferimento al file **fibo.scala** per un esempio pienamente funzionante che usi **myFibo**. Il ciclo **for** usato in **fibo.scala** può avere diverse forme; in questo caso viene usato un range. Altre forme includono l'uso di filtri e la parola chiave **yield**. Il problema con l'implementazione ricorsiva precedente è che diventa estremamente lenta nel calcolare grandi numeri di Fibonacci e può generare eccezioni di stack overflow, dal momento che ogni volta che una funzione chiama un'altra funzione (anche sé stessa) lo stato di entrambe le funzioni dev'essere salvato fino al termine della loro esecuzione. Provate a calcolare il cinquantesimo numero di Fibonacci e capirete. La **Fig 2** mostra il numero di calcoli necessario per trovare il quinto numero di Fibonacci usando la ricorsione pura. Ogni calcolo necessita di diverse chiamate alla funzione **myFibo()**. È quindi d'uopo imparare a calcolare i numeri di Fibonacci più velocemente. Il codice Scala per la funzione **fastFibo()** è:

```
def fastFibo( n : Int) : Int = {
        def fibo_tail(n: Int, a:Int, b:Int): Int = n match {
                case 0 => a
                case _ => fibo_tail(n-1, b, a+b)
        }
        return fibo_tail(n, 0, 1)
}
```

Questo metodo usa la ricorsione in coda, che torna utile dal momento che le chiamate in coda possono essere implementate senza aggiungere un nuovo frame allo stack di chiamate esistenti. Il calcolo del 40esimo numero di Fibonacci è praticamente istantaneo. La funzione **fibo_tail** prende tre parametri interi in input e ritorna un intero. La **Fig 2** mostra anche il numero di calcoli necessari per



› Fig 2: Calcolare il quinto numero di Fibonacci usando un'implementazione a pura ricorsione e una con ricorsione in coda

trovare il quinto numero di Fibonacci usando la ricorsione in coda. C'è una sola chiamata a **fastFibo()**. La differenza nella chiamata di funzione è comunque notevole, malgrado il fatto che sia necessario un numero minore di calcoli per calcolare il numero desiderato.

Implementare Bubble Sort

Questa sezione vi mostrerà come implementare il Bubble Sort in Scala usando il tradizionale metodo iterativo, dal momento che Scala supporta anche la programmazione iterativa. Questo rappresenta un grosso vantaggio perché potete imparare a programmare gradualmente usando il modo funzionale continuando a scrivere le parti necessarie nella maniera che conoscete già. Il codice completo per la funzione che implementa Bubble Sort è il seguente:

```
def bubbleSort(input: Array[Int]): Array[Int] = {
        val limit = input.size - 1
        for (a <- 1 to limit) {
                for (b <- limit to a by -1) {
                        if (input(b) < input(b - 1)) {
                                val x = input(b)
                                input(b) = input(b - 1)
                                input(b - 1) = x
                        }
                }
        }
        input
}
```

La funzione **bubbleSort()** prende un array di interi come input. Il suo output è sempre un array di interi. Se siete familiari con l'algoritmo Bubble Sort non avrete problemi »

Come Scala e Java sono relazionati

Scala e Java sono in relazione perché Scala gira sulla Java Virtual Machine. Le classi Scala, quindi, sono in definitiva delle classi JVM. Come risultato potete programmare oggetti Java, chiamarne i metodi ed ereditare classi Java trasparentemente da Scala. Come potete immaginare, le librerie Java possono essere usate direttamente nel codice Scala. In più, il codice Java può referenziare facilmente classi e oggetti Scala. La **Fig 4** mostra più informazioni, oltre al codice necessario. Anche se non siete grandi fan di Java, poter chiamare codice Scala dai vostri programmi Java è un enorme vantaggio; se non vi piace Java, non usate questa caratteristica!
In ogni caso Scala supporta molte caratteristiche non supportate da Java, inclusi la type inference, immutabilità, lazy evaluation e pattern matching. Scala ha anche un sistema di tipi avanzato che supporta tipi di dato algebrico, covarianza e controvarianza, tipi higher-order e anonimi. Altre caratteristiche di Scala non presenti in Java includono l'overload degli operatori, parametri opzionali, parametri named, stringhe raw e eccezioni non controllate. Scala è progettato per migliorare l'esperienza di programmazione Java e per essere slegato dai limiti di Java. L'unica cosa che Scala non può sistemare è il fatto che eseguire un programma che usi la JVM consumi più risorse dell'esecuzione di un mero file binario.

» a comprendere l'implementazione Scala. Dal momento che ogni nuova variabile Scala dev'essere definita con la parola chiave **var**, dovete definire una nuova variabile array usando **var** davanti al suo nome e inizializzarla con il valore di ritorno di una funzione usando il codice **var myArray = createArray(10)**. Una variabile dichiarata con **val** in luogo di **var** non potrà cambiare successivamente il proprio valore. La funzione **scala.util.Random** viene usata in **createArray()** per popolare l'array con numeri casuali prima di ordinarli. Per il programma completo fate riferimento al file **bubble.scala**.

Argomenti da riga di comando

In questa sezione vedrete come usare Scala per processare gli argomenti da riga di comando. Sapete già dalla definizione della funzione **main()** che gli argomenti da riga di comando vengono passati usando l'array **args**. Ora vedrete come recuperarli e manipolarli nei vostri programmi Scala. Il modo più semplice di stampare gli argomenti forniti a riga di comando è usando la riga di codice **args foreach println** all'interno del vostro programma. Il comando usa l'istruzione foreach per recuperare tutti gli elementi dall'array args e stamparli a schermo usando il comando **println**. Il codice seguente controlla tutti gli argomenti da riga di comando e stampa quelli che sono interi positivi:

```
        for ( x <- args ) {
                if (isAllDigits(x)) {
                        println(x + " is a positive
integer!")
                }
        }
 def isAllDigits(x: String) = x.map(Character.isDigit(_)).
reduce(_&&_)
```

Tutto il lavoro sporco è fatto dalla funzione **isAllDigits()** che sfrutta **map** e **reduce** per controllare se la stringa fornita è composta interamente da cifre. Fate riferimento al file **args. scala** per vedere tutto il codice Scala. Dal momento che args è un normale array, potete controllare il numero corretto di argomenti della riga di comando usando il codice seguente:

```
 if ( args.size != 2 ) {
        println("Dovreste fornire esattamente due argomenti
a riga di comando.")
        System.exit(1)
 }
```

La chiamata **System.exit(1)** viene usata per terminare immediatamente il programma Scala. Essere in grado di recuperare l'input dall'utente è molto utile. Questa sezione vi mostrerà anche una tecnica per farlo in Scala. Il codice seguente chiede all'utente una stringa in input e la memorizza in una variabile chiamata **userName**:

```
 val userName = readLine("Inserisci il tuo nome: ")
```

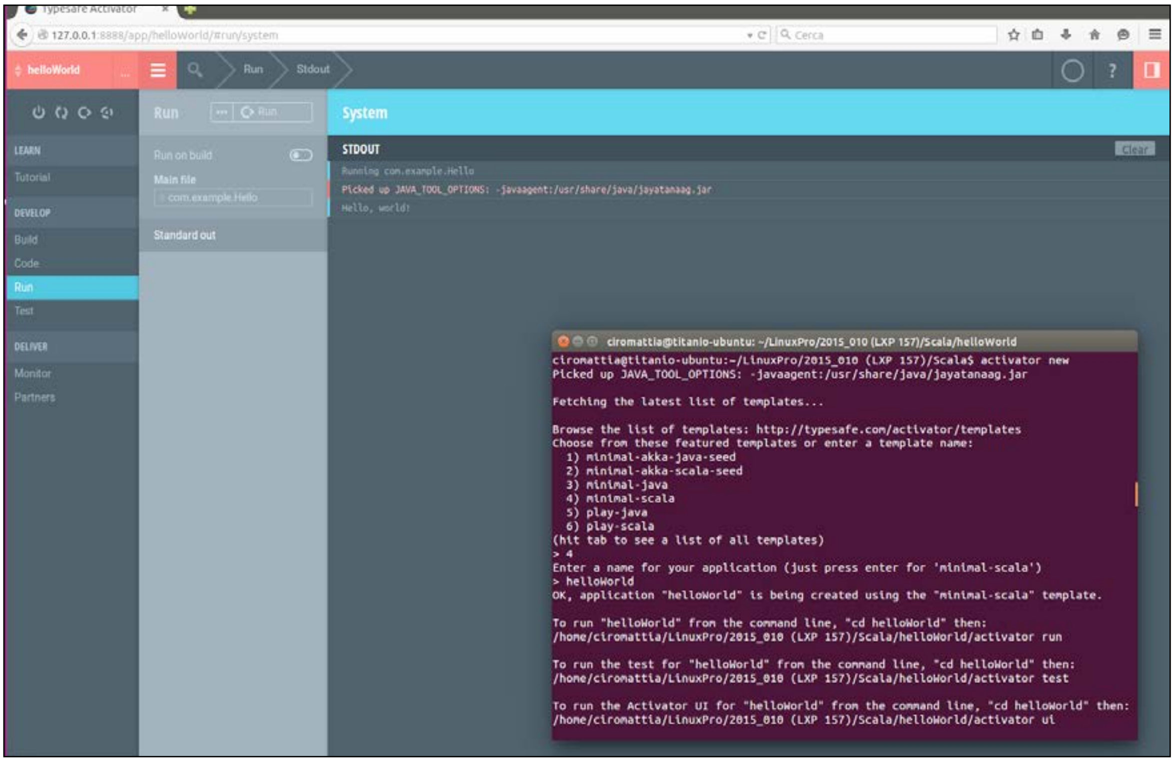
La funzione **readInt()** vi permette di leggere un intero, ma dovrete usare **println()** per far sapere all'utente che deve scrivere qualcosa. Scrivere un valore non intero a **readInt()** causerà un'eccezione.

File I/O in Scala

È il momento di costruire un piccolo programma che legga un file di testo riga per riga, inserisce i numeri di riga in testa a ognuna e salva l'output in un nuovo file di testo. Entrambi i file vengono passati come argomenti a riga di comando.
Potete iterare sui contenuti di un file riga per riga, come segue:

```
 import scala.io.Source
 for (line <- Source.fromFile(input).getLines()) {
        println(line)
 }
```

La riga **import scala.io.Source** istruisce Scala di caricare la classe **scala.io.Source**, necessaria per leggere un file di testo. Qui viene usato un ciclo for, ma potete usare anche un ciclo foreach se vi piace di più. Potete scrivere in un file di testo utilizzando la funzione seguente:



› **Fig 3: Questo è l'output di una piccola applicazione Web sviluppata in Scala usando il framework *Play***

Pattern matching in Scala

Come succede per qualsiasi linguaggio di programmazione moderno, il pattern matching gioca un ruolo fondamentale in Scala. Quindi, nello sviluppare programmi in Scala, il pattern matching può far sì che il vostro codice sia elegante, intelligente e facile da leggere dal momento che offre all'utente un modo di programmare strutture alternative basate sulle vostre necessità e su quelle dei problemi che state cercando di risolvere. Il codice seguente mostra un modo di individuare tutti gli interi:

```
def testForInteger(x: Int): String = x match {
  case 1 => "uno"
  case 3 => "tre"
  case _ => "ALTRO"
}
```

Il pattern matching non dev'essere semplicistico, come mostra il codice seguente:

```
def whatIsIt(x: Any): String = x match {
  case i:Int => "Un intero: " + i
  case d:Double => "Un double: " + d
  case s:String => "Una stringa: " + s
  case _ => "Sconosciuto!"
}
```

L'ordine dei casi nel pattern matching è importante perché solo il primo match verrà eseguito. Se avete un errore logico del genere nel vostro codice Scala, potreste ricevere un warning. Provare e implementare una semplice versione di grep usando Scala sarebbe un ottimo esercizio! Non provate a implementare qualsiasi opzione di grep in una volta sola, assicuratevi sempre di avere un programma stabile senza bug.

```
def appendFile(fileName: String, line: String) = {
        val fileToWrite = new FileWriter(fileName, true)
        try {
                fileToWrite.write(line + "\n")
        }
        finally fileToWrite.close()
}
```

Dovrete usare codice Java per scrivere su un file di testo, dal momento che Scala non dispone di una libreria di I/O stabile. Dal momento che la funzione **fileToWrite.write()** non inserisce automaticamente il carattere newline alla fine di ogni riga, dovete inserirlo manualmente. Se il file in output non esiste sarà creato automaticamente. Se già esiste, tutte le nuove righe saranno aggiunte in coda. Per vedere il codice Scala in tutta la sua gloria, fate riferimento a **file.scala** all'interno del quale capirete anche come controllare se un file esiste già prima di leggerlo, usando del codice Java!

Programmazione funzionale

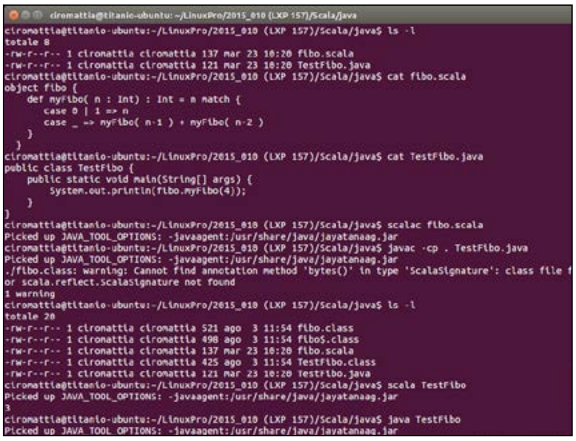
Potreste aver notato che l'ultimo trend nei linguaggi di programmazione è la programmazione funzionale. Scala non è un'eccezione a questa regola non ufficiale e qui lo vedrete in dettaglio con due esempi. Il primo è l'utilizzo delle funzioni anonime (funzioni letterali) mentre il secondo usa le funzioni come variabili. Il codice seguente dichiara una lista e ne stampa quindi i numeri dispari all'interno, usando una funzione anonima:

```
scala> val myList = List(1, 3, 4, 5, 6, 0, 1)
myList: List[Int] = List(1, 3, 4, 5, 6, 0, 1)
scala> val odds = myList.filter((i: Int) => i % 2 == 1)
odds: List[Int] = List(1, 3, 5, 1)
```

Il codice per la funzione letterale usato per filtrare l'input è **(i: Int) => i % 2 == 1**. Potete pensare al simbolo **=>** come un trasformatore, dal momento che l'espressione trasforma quello che sta alla sinistra dell'operatore usando l'algoritmo sul lato destro. Il vantaggio è che non dovete implementare una funzione separata per fare il lavoro! Potete filtrare un array allo stesso modo. Potete anche assegnare una funzione anonima a una variabile: **var mySquare = ((i:Int) => i * i)** definisce una variabile chiamata **mySquare** che ritorna il quadrato di un intero. Da qui potete usare **mySquare()** come fosse una normale funzione. Potete vedere entrambi i casi in azione nel file **funprog.scala**.

Applicazioni Web

Al giorno d'oggi tutti i linguaggi di programmazione moderni hanno un modo di sviluppare applicazioni Web: Scala non fa



› Fig 4: Questo è il modo di usare, compilare e lanciare codice Scala dai vostri programmi Java

eccezione! Questa sezione vi mostrerà come programmare una piccola applicazione con l'aiuto di Scala. I due framework Scala più conosciuti sono **Lift** (**http://liftweb.net**) e **Play** (**https://www.playframework.com**), basato su Java NIO e altamente scalabile. La più grande differenza tra i due è che Lift è *stateful* mentre Play è *stateless*, analogamente a Rails e Django. Il sito Web di Play contiene informazioni su come installarlo nella vostra macchina. A seguito dell'installazione dovrete usare il comando **activator new** per generare un nuovo progetto. Una volta fatto questo, scegliete il tipo di progetto minimal-scala. Il nome sarà helloWorld, e verrà generata una nuova directory con lo stesso nome. Eseguire **./activator ui** dalla directory del progetto inizia molte cose e produce molto output. La parte più interessante dell'output è questa:

```
Play server process ID is 20053
[info] play - Application started (Prod)
[info] play - Listening for HTTP on /127.0.0.1:8888
[info] a.e.s.Slf4jLogger - Slf4jLogger started
[info] application - onStop received closing down the app
```

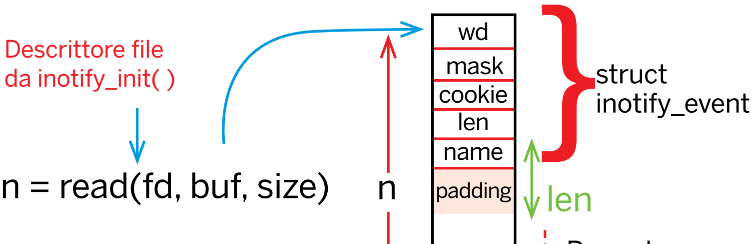
Il file con il codice Scala è **src/main/scala/com/example/Hello.scala**. Potete vedere l'intera applicazione di default puntando il vostro browser su **http://localhost:8888**. La Fig 3 mostra l'output dell'applicazione Web e tutto l'ambiente. Approfondire ulteriormente lo sviluppo di applicazioni con il framework Play è, tuttavia, oltre gli scopi di questo articolo. Un ottimo modo di imparare Scala è implementarvi dei tradizionali strumenti UNIX. Potete cominciare con alcuni semplici tool e progredire con quelli più complessi, come **ls**, **rm**, **df** e **mv**, mano a mano che la vostra esperienza aumenta. LXP

System coding: inotify & getopts

In questa terza puntata scoprirete come seguire le modifiche al filesystem e processare gli argomenti da riga di comando

Ecco uno scenario per voi: immaginate di stare scorrendo i contenuti di una directory con un file manager grafico. Aggiungete un nuovo file alla directory, magari copiandolo da riga di comando, ed ecco che il file manager aggiorna la vista istantaneamente per includere il nuovo file. Come ha rilevato che il file è comparso? Beh, potrebbe fare un poll per modifiche, elencando ripetutamente i contenuti della directory e controllando se ci sono nuovi file. Il polling però non è un uso efficiente delle risorse. Fortunatamente, esiste un meccanismo nel kernel Linux esattamente per questo scopo: notificare a un programma quando si verificano modifiche nel filesystem. Si chiama **inotify**. Ecco la panoramica. Per prima cosa, create un'istanza inotify e ottenete un descrittore di file, esattamente come succederebbe con una open() su un file. Poi aggiungete dei 'watch'. Banalmente, un watch dice "vorrei tener d'occhio questo file, o questa directory, per questa e quest'altra modifica". Quindi restate in attesa di messaggi di notifica sulle modifiche, che leggete dal descrittore. Potete aggiungere tanti watch quanti volete, e potete controllare file individuali o directory, ma non è ricorsivo: un watch su una directory non controlla automaticamente le sottodirectory. Ecco subito un esempio. Questo piccolo programma controlla due directory (**home** e **/etc**) e stampa un messaggio se un file viene creato o cancellato:

```
1 /* Controlla le directory per modifiche usando inotify */
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <sys/inotify.h>
6 #include <limits.h>
7
8 /* Un buffer grande abbastanza per contenere 100 eventi */
9 #define BUFSIZE (100 * (sizeof(struct inotify_event) +
NAME_MAX + 1))
10
11 void main()
12 {
13          int notifyfd, watchfd;
14          char eventbuf[BUFSIZE]; /* Gli eventi saranno letti
qui */
15          int n;
16          char *p;
17          struct inotify_event *event;
18
19          notifyfd = inotify_init(); /* Senza controllo d'errore */
20
21          watchfd = inotify_add_watch(notifyfd, "/home/
chris", IN_CREATE | IN_DELETE);
22 watchfd = inotify_add_watch(notifyfd, "/etc", IN_ CREATE
| IN_DELETE);
23
24          /* I watch sono impostati, leggete il flusso eventi */
25
26          while(1) {
27                   n = read(notifyfd, eventbuf, BUFSIZE);
28
29                   /* Ciclare sugli eventi e riportarli */
30
31                   for (p = eventbuf; p < eventbuf + n;) {
32                            event = (struct inotify_event *) p;
33                            p += sizeof(struct inotify_
event) + event_len;
34                            /* Mostra l'evento */
35                            if (event->mask & IN_
CREATE) printf("%s creato\n", event->name);
36                            if (event->mask & IN_
DELETE) printf("%s cancellato\n", event->name);
37                   }
38          }
39 }
```



› I messaggi evento letti dal descrittore di *inotify* contengono un nome di lunghezza variabile e sono quindi difficili da leggere nel buffer

La parte interessante comincia alla riga 19, dove viene inizializzato il flusso di eventi ottenendo un descrittore di file. Alle righe 21 e 22 vengono aggiunti i watch nelle directory che vi interessano. Le macro **IN_CREATE** e **IN_DELETE** sono valori a singolo bit intesi per essere usati come maschera per

specificare a quali eventi siete interessati. Ci sono un paio di dozzine di tipologie, elencati per esteso nella pagina man di inotify. Una volta definiti tutti i watch, entrate in un ciclo (riga 27) per leggere gli eventi di modifica nel flusso. Vi vengono ritornati uno o più messaggi evento, definiti con la struttura inotify_event. La struttura ha un campo di lunghezza variabile alla fine per contenere il nome del file che ha scatenato l'evento (figura in basso a sinistra). Questo rende un po' difficile il ciclo sugli eventi nel buffer, richiedendo un po' di aritmetica dei puntatori esplicita (riga 33). Per indicare quale evento è successo, il campo della maschera in inotify_event usa gli stessi valori di bit usati in precedenza per specificare a quali eventi eravate interessati. Alle righe 35 e 36 controllate i due eventi che state controllando e stampate un messaggio adatto in ogni caso. Potete facilmente immaginare che oltre all'applicazione in un file manager grafico, inotify può essere usato per creare un log di modifiche a file importanti.

Controllare gli argomenti da riga di comando

Cambio di argomento. Avete familiarità, naturalmente, con il passare argomenti a un programma da riga di comando. Se usate gli script di shell, sapete che tali argomenti sono accessibili tramite le variabili $1, $2 e così via. Ma come li recuperate da un programma C? Ebbene, questi sono contenuti in due argomenti passati nella funzione main(). Naturalmente potete chiamarli come più vi piace, ma dareste uno schiaffo alle convenzioni se non li chiamaste **argc** e **argv**. Il primo argomento, argc, vi ritorna semplicemente il numero di argomenti presenti. Il secondo, argv, punta a un array di puntatori, che a loro volta puntano alle stringhe che contengono gli argomenti effettivi. Notate che il nome del programma lanciato è incluso nella lista (accessibile tramite argv[0]) e incluso nel conteggio di argc (figura sotto). Potete quindi attraversare gli argomenti da riga di comando con un semplice ciclo:

```
int main(int argc, char* argv[])
{
        int i;
        for (i=0; i<argc; i++)
                printf("%s\n", argv[i]);
}
```

Spesso vorrete controllare che l'utente abbia fornito un numero appropriato di argomenti. Ecco un esempio di programma che si aspetta esattamente un nome di file:

```
int main(int argc, char* argv[])
{
        char *filename;
        if (argc != 2) {
                fprintf(stderr, "Utilizzo: %s file\n",
argv[0]);
                exit(1);
        }
        filename = argv[1];
        /* ... */
}
```

Notate l'uso di **argv[0]** per includere il nome del programma nel messaggio d'aiuto. Occasionalmente troverete programmi che esaminano argv[0] e determinano il loro comportamento a seconda di come sono stati invocati. Questa pratica era in voga anni fa e ora è diminuita, ma se controllate unzip e zipinfo vedrete che fanno riferimento allo stesso eseguibile. Se lanciate:

```
$ find /bin /usr/bin -type l -ls
```

scoprirete diversi link simbolici a programmi che (presumibilmente) usano la stessa tecnica.

Usare getopts() per processare

Questa cosa di argv non sembra male, sempre se vi sentite a vostro agio nel gestire un array di puntatori. Può diventare però arduo. Considerate il comando:

```
ls -a -l foo bar
```

Convenzionalmente, gli argomenti che iniziano con un '-' sono trattati come opzioni. Naturalmente potete identificarli con un po' di gestione delle stringhe. Nessun problema. A questo punto realizzate che dovete gestire anche la sintassi equivalente:

```
ls -al foo bar
```

usata da tutti, e anche

```
ls foo bar -a -l
```

difficilmente usata da qualcuno, ma comunque legittima. Altra gestione delle stringhe. E qui il gioco si fa duro: le opzioni sono in fondo, ma dovete processarle per prime. Considerate anche:

```
gcc -o demo demo.c
```

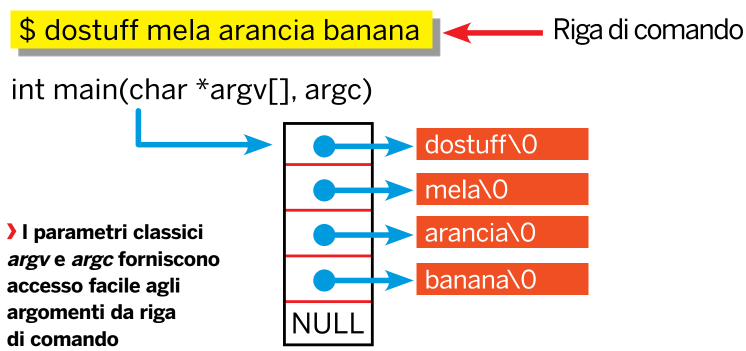
dove l'opzione **-o** è accoppiata con l'argomento **demo** per specificare il nome del file in output. Sì, sì, potete aumentare la gestione delle stringhe, ma diventano un sacco di casi da gestire e, siate onesti, C non è un granché per trattare le stringhe. Non è però tutto morte e disperazione. Esiste una funzione di libreria chiamata **getopts()** che vi alleverà dal grosso del lavoro per processare gli argomenti da riga di comando. I dettagli sono piuttosto confusi, quindi accontentatevi della panoramica: chiamate getopts() ripetutamente sul vostro elenco in argv e vi ritornerà la lettera dell'opzione, una per una. Nel caso l'opzione accetti un argomento, vi ritornerà anche quello. Mescolerà inoltre l'elenco in modo da farvi trovare alla fine tutti gli argomenti non-opzione e vi dirà dove iniziano:

```
1 /* Processo argomenti da riga di comando con getopt() */
2 /* Uso: argdemo -a -b -n 400 -t purple */
3
4 #include <unistd.h>
5 #include <stdio.h>
6 #include <string.h>
7
8 void main(int argc, char *argv[])
9 {
10      int c;
11      int aflag = 0;
```

»

Tip

La variabile d'ambiente POSIXLY_CORRECT viene usata per forzare la correttezza POSIX ed è usata principalmente in situazioni "dove POSIX e il buonsenso non vanno d'accordo". Come esempio, se POSIXLY_CORRECT è impostata, getopts() smetterà di controllare le opzioni negli argomenti appena incontrerà il primo argomento non-opzione.



› **I parametri classici *argv* e *argc* forniscono accesso facile agli argomenti da riga di comando**

Tip

Se mai doveste aver bisogno di passare un argomento che comincia con '-' ma non dev'essere preso come opzione, l'argomento '--' terminerà il controllo delle opzioni per la restante riga di comando. Per esempio: **$argdemo -a -- -b** riconoscerà -a come opzione ma tratterà -b come argomento non-opzione.

```
12        int bflag = 0;
13        int numoption = 0;
14        char txtoption[200] = "default";
15
16        opterr = 0;
17
18        while ( (c = getopt(argc, argv, "abn:t:")) != EOF) {
19                switch (c) {
20                case 'a':
21                        aflag = 1;
22                        break;
23                case 'b':
24                        bflag = 1;
25                        break;
26                case 'n':
27                        numoption = atoi(optarg);
28                        break;
29                case 't':
30                        strcpy(txtoption, optarg);
31                        break;
32                case '?':
33                        fprintf(stderr, "opzione non 
valida: -%c\n", optopt);
34                }
35        }
36
37        if (aflag) printf("Opzione a impostata\n");
38        if (bflag) printf("Opzione b impostata\n");
39        printf("Il valore dell'opzione numerica è %d\n", 
numoption);
40        printf("Il valore dell'opzione testo è %s\n", 
txtoption);
41
42        argv += optind; /* Sposta il puntatore dopo le 
opzioni */
43        argc -= optind;
44
45        while (argc--) { /* Stampa gli argomenti non-opzione 
*/
46                printf("arg: %s\n", *argv++);
47        }
48 }
```

Il commento in riga 2 mostra la sintassi del comando che volete lanciare, con **-a** e **-b** semplici opzioni booleane, **-n** opzione numerica e **-t** opzione testuale. Il capo del ciclo getopt è alla riga 18. Vi passate argc e argv, e il terzo argomento ("abn:t:" nell'esempio) indica a getopt che a, b, n e t sono opzioni valide e che **-n** e **-t** si aspettano i propri argomenti. getopt() spazzola tutta la riga di comando ritornando la lettera dell'opzione mano a mano che la rileva, e pone gli argomenti non-opzione in fondo all'array argv. Le opzioni individuali sono processate nei case dell'istruzione switch (righe 19-31). Le opzioni **-a** e **-b** impostano semplicemente le flag booleane aflag e bflag (per essere utilizzate più avanti nel programma, presumibilmente). Nel caso delle opzioni **-n** e **-t** getopt() mette l'argomento seguente nella variabile globale **optarg**, che viene estratta come numero (riga 27) o come stringa (riga 30). L'ultimo case (riga 32) viene attivato nel caso venga rilevata un'opzione inaspettata (per esempio "-x"). Quando getopt() non può trovare ulteriori opzioni, ritorna EOF e si esce dal ciclo. A questo punto, tutti gli argomenti non opzione sono alla fine dell'array argv e la variabile globale **optind** è l'indice del primo. Aggiungete tale valore a argc e argv (righe 42, 43) prima di ciclare sugli argomenti rimanenti (righe 45, 46). Notate che qui l'incremento del puntatore è fatto all'interno della chiamata printf(), il che è più in stile C, invece dell'approccio Fortran-esco nel ciclare su un indice intero adottato prima.
Ecco il programma in azione:

```
$ argdemo -a mela arancia
Opzione a impostata
Il valore dell'opzione numerica è 0
Il valore dell'opzione testo è default
arg: mela
arg: arancia
$ argdemo -n 500 -ab
Opzione a impostata
Opzione b impostata
Il valore dell'opzione numerica è 500
Il valore dell'opzione testo è default
$ argdemo banana pera -t verde
Il valore dell'opzione numerica è 0
Il valore dell'opzione testo è verde
arg: banana
arg: pera
```

Potete comprendere da soli questi tre esempi. Notate che -ab funziona allo stesso modo di -a -b, e l'ultimo esempio mostra che le opzioni sono riconosciute correttamente anche se vengono dopo le non-opzioni.

Processare opzioni 'lunghe'

Sembra che le opzioni 'lunghe' siano state introdotte in Linux dalla gente di GNU. Alcune hanno equivalenti tradizionali (a lettera singola, per esempio **ls --all** è equivalente a **ls -a**) mentre altre no (per esempio **ls --color**). Probabilmente la necessità dietro alle opzioni lunghe è stata che molti comandi stavano esaurendo le lettere (GNU C Compiler, per esempio, ha più di 400 opzioni). Negli script, le opzioni lunghe possono essere più comprensibili delle loro criptiche cugine a una lettera. Potete processare le opzioni lunghe usando **getopt_long()** nella libreria C GNU, ma è un po' tedioso. Per ogni opzione, dovrete costruire una struttura opzione contenente il nome dell'opzione stessa e alcune flag specificando se accetta un argomento associato e come dev'essere trattata. Quindi dovrete passare un array di tali strutture. Il modulo **parseargs** di Python semplifica questa cosa permettendovi di gestire le opzioni lunghe come le altre, per esempio:

```
p.add_argument("-a", action="store_
true", dest="aflag")
p.add_argument("--all", action="store_
true", dest="aflag")
```

Processare gli argomenti in Python

Python fornisce una lista chiamata s**ys.argv** che serve allo stesso scopo di argv nel programma C. Ecco la versione Python del programma mostrato poc'anzi, che si aspetta un nome di file:

```
import sys
if (len(sys.argv) != 2):
        print("uso", sys.argv[0], "file")
        raise SystemExit(1)
filename = sys.argv[1]
# e via dicendo
```

Notate che non c'è equivalente ad argc. Non è necessario che ci sia, dal momento che in Python un elenco come sys.argv sa quanto lungo è. Python ha anche un modulo chiamato **getopts** che funziona analogamente a getopts() nella libreria C standard appena esaminata. Tuttavia c'è anche un modulo chiamato argparse che fornisce un

supporto a un livello sostanzialmente più alto. Ecco il programma mostrato prima (più o meno) riscritto usando il modulo **argparse** di Python:

```
1 #!/usr/bin/python3
2 import argparse
3
4 p = argparse.ArgumentParser()
5
6 p.add_argument("-a", action="store_true", dest="aflag")
7 p.add_argument("-b", action="store_true", dest="bflag")
8 p.add__argument("-n", action="store", type=int,
dest="numoption")
9 p.add_argument("-t", action="store", dest="txtoption")
10 p.add_argument('files', nargs='*')
11
12 p.set_defaults(aflag=False, bflag=False,
13                          numoption=0, txtoption="default")
14
15 args = p.parse_args()
16
17 if (args.aflag): print("Opzione a impostata")
18 if (args.bflag): print("Opzione b impostata")
19 print("Il valore dell'opzione numerica è ", args.numoption)
20 print("Il valore dell'opzione testo è ", args.txtoption)
21
22 print(args.files)
```

Ecco un piccolo tour. Create un ArgumentParser alla riga 4; sarà quello che farà tutto il lavoro sporco. Le righe 6-10 informano il parser sulle opzioni valide. Per esempio, la riga 8 indica al parser di aspettarsi un'opzione **-n** e memorizzare l'intero associato in **numoption**. La chiamata parse_args() alla riga 15 fa succedere tutta la magia: vi viene ritornato un oggetto di namespace (args) all'interno del quale trovate le variabili contenenti i valori degli argomenti rilevati da riga di comando. Nelle righe 17-20 le controllate e quindi stampate tutti gli argomenti non-opzione alla riga 21. Come design, il programma funziona in modo simile alla versione C vista in precedenza. Il vantaggio di ArgumentParser diventa visibile se indicate deliberatamente un'opzione non valida:

```
$ ./argdemo.py -x
usage: argdemo.py [-h] [-a] [-b] [-n NUMOPTION] [-t
TXTOPTION] [files [files ...]]
argdemo.py: error: unrecognized arguments: -x
```

Magia! Senza ulteriore codice, Python ha generato un messaggio di utilizzo dettagliato e riportato l'incorrettezza della opzione -x. Ottenete anche, in più, le opzioni -h e --help, e potete migliorarne l'output assegnando testi descrittivi a ogni argomento. È consigliato leggere la documentazione di Python su argparse, vi convincerà del fatto che Python sia ben a più alto livello di C!

L'environment

Per chiudere questa puntata, ecco un'introduzione all'ambiente (environment). Ogni processo ne ha uno, ed essenzialmente è un elenco di stringhe, ognuna nella forma

```
NOME=VALORE
```

Gli elementi individuali sono chiamati variabili d'ambiente e convenzionalmente sono in maiuscolo. Potreste già esservi familiari nell'utilizzo nella shell. Potete chiedere alla shell di elencare tutto l'ambiente con il comando **env**, e potete aggiungere variabili all'ambiente con il comando export, per esempio:

```
$ FOO=BAR
$ export FOO
```

oppure in una riga con

```
$ export FOO=BAR
```

Sia chiaro, tuttavia, che assegnare un valore a una variabile shell NON la aggiunge all'ambiente. Potete accedere all'ambiente nei vostri programmi in due modi. Per prima cosa, la variabile globale **environ** punta a un array di puntatori che a loro volta puntano a stringhe individuali, analogamente ad argv (vedete la figura in basso a destra). Potete quindi elencare tutto l'ambiente in questo modo:

```
extern char **environ;

main()
{
        char **p;
        for (p=environ; *p != NULL; p++)
                printf("%s\n", *p);
}
```

Solitamente vorrete solo ottenere il valore di una variabile specifica, ottenibile più semplicemente con getenv():

```
getenv("DISPLAY")
```

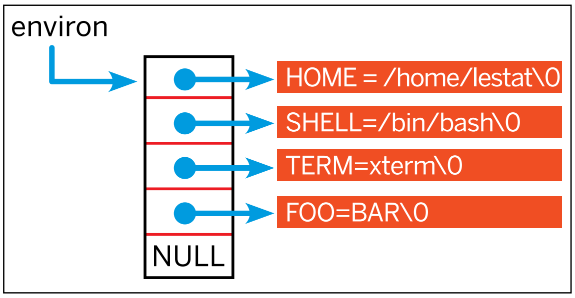
ritorna un puntatore al valore della variabile d'ambiente DISPLAY, o NULL se non esiste la variabile stessa. Altre chiamate di sistema per manipolare l'ambiente includono setenv() e unsetenv(), equivalenti ai comandi Bash integrati **export** e **unset**, e clearenv() che cancella tutto l'ambiente. Ora, il punto dell'ambiente è che solitamente viene trasmesso ai processi figli. Questo lo rende un ottimo meccanismo per trasmettere opzioni di configurazione che volete siano viste da tutti i programmi che lanciate. Notate che questa eredità funziona in un senso solo: le modifiche fatte all'ambiente di un processo figlio non si propagano all'indietro al padre, né le modifiche fatte all'ambiente del padre dopo che l'ambiente è stato propagato al figlio. Le variabili d'ambiente sono tipicamente usate per passare piccoli pezzi di configurazione a un programma. Molti programmi interrogano variabili specifiche, e lo stesso fanno alcune routine nella libreria C standard. Per esempio il programma **crontab**, che vi permette di modificare il file **crontab** e ne controlla la sintassi, usa la variabile d'ambiente EDITOR per decidere qual è il vostro editor preferito. Quindi i comandi:

```
$ export EDITOR=nano
$ crontab -e
```

vi farà modificare il crontab con nano. Potete lanciarlo così:

```
$ EDITOR=nano crontab -e
```

che eseguirà crontab con la variabile d'ambiente EDITOR definita, ma non modificherà l'ambiente della shell. Come altro esempio, la variabile d'ambiente **TZ** (time zone) è utilizzata da diverse routine nella libreria standard che converte un orario in locale. Per i dettagli dovrete aspettare la prossima puntata! LXP



› Analogamente ad *argv*, l'environment è reso disponibile con un array di stringhe terminate dal carattere *null*

Tip

Potete esaminare l'environment di ogni processo se ne conoscete il process ID guardando il file ***/proc/PID/environ*** dove 'PID' è il process ID. Gli elementi sono terminati da caratteri null, non newline, quindi sono difficili da leggere.

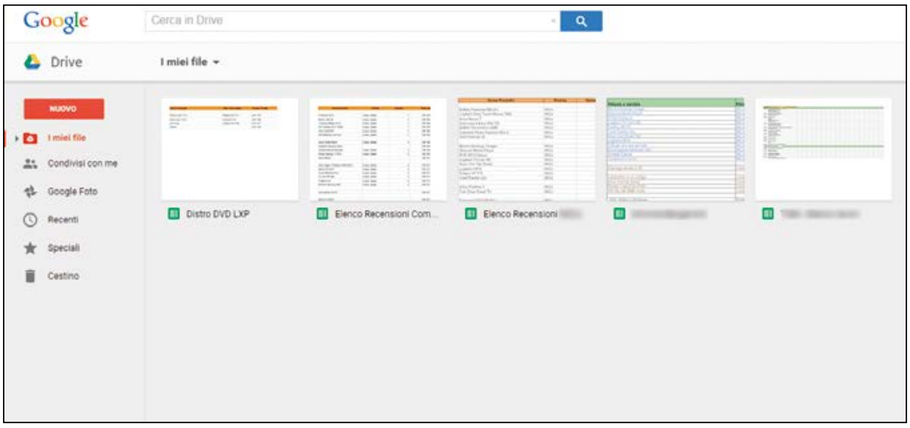
Domande & Risposte

Avete qualche domanda sull'Open Source? Scriveteci a **mailserver@linuxpro.it** per ottenere una risposta

1 Google Drive per Linux

Grazie all'acquisto di un Chromebook, ho ottenuto ben 100 GB di spazio gratuito su Google Drive. Utilizzando la mia distro Linux, però, posso caricare e gestire i file solo dal browser Web. Mi piacerebbe invece farlo direttamente dal desktop.

Purtroppo, rispetto a Windows, Google non si è ancora deciso a fornire un client Linux per la gestione di Drive direttamente da desktop. Tuttavia non è necessario disperarsi, perché esistono valide alternative. Una di queste, per esempio, è **GSync** (**http://bit.ly/Gsync4GoogleDrive**) che emula le opzioni di comando di rsync. In questo modo, chi utilizza già questo strumento non dovrà imparare niente di nuovo. Si tratta di un programma Python da installare usando **pip**. Se non lo hai mai usato prima d'ora, potrebbe essere necessario installarlo. Prima di tutto



› ***Google Drive* è un'ottima piattaforma Cloud, peccato che il colosso di Mountain View non abbia ancora pensato a fornire un client desktop per Linux**

assicurati di avere il pacchetto **python-setuptools** (la maggior parte delle distro ne dispone per impostazione predefinita):

```
sudo easy_install pip
```

Adesso puoi installare **gsync** con:

```
sudo pip install gsync
```

Pip si occuperà quindi di recuperare tutte le dipendenze relative a gsync. Al primo utilizzo di gsync, potresti trovarti ad avere a che fare con un bug che impedisce l'upload dei file. In questo caso, devi aprire come root il file **usr/lib64/python2.7/site-packages/libgsync/drive/__init__.py** in un editor di testo qualsiasi, quindi andare alla riga 744 dove si trovano queste istruzioni:

```
for k, v in properties.iteritems():
    body[k] = _Drive.utf8(v)
```

cambiale poi con:

```
for k, v in properties.iteritems():
    if v is not None:
      body[k] = _Drive.utf8(v)
```

Fai molta attenzione ai rientri che, come sempre quando parliamo di Python, rivestono un ruolo importante.

Adesso, esegui:

```
sudo gsync
```

Ricorda che è sufficiente eseguire una sola volta questo comando, così da compilare la modifica appena apportata. Adesso è possibile sincronizzare una directory locale con Google Drive tramite l'istruzione:

```
rsync -rv /local/directory drive://directory
```

La prima volta che si esegue questo comando, gsync creerà una URL piuttosto lunga. Copiala nel browser e Google ti chiederà di accedere al tuo account. Una volta fatto, verrà fornito un codice che deve essere copiato e incollato nel terminale. Solo a questo punto gsync inizierà a copiare i file dal tuo account

Riflettori su...

Sapreste indicarmi una serie di distro firewall?

Esistono varie distribuzioni il cui unico scopo è proteggere la rete dove sono installate. Una delle migliori è **IPCop**, disponibile per il download alla pagina ufficiale **www.ipcop.org**. Nel momento in cui scriviamo, la versione più aggiornata è la 2.1.8. La distro dispone di un'ottima documentazione che si snoda tra forum e mailing list. Come alternativa possiamo poi citare **Untangle NF Firewall**, una distro basata su Debian che, oltre a prevenire eventuali intrusioni, è in grado di gestire VPN, SSL e molto altro. La trovate su **www.untangle.com**. Infine, segnaliamo anche **Endian Firewall**, **www.endian.com**, una distro *Unified Threat Management* (UTM) basata su Red Hat Enterprise Linux che può vantare componenti al 100% Open Source.

Terminale e superutente

Nelle distro Linux ci sono diversi modi per applicare un comando a una data operazione. La maggior parte delle volte, infatti, potete procedere sia utilizzando degli strumenti grafici sia la potenza della riga di comando. In quest'ultimo caso, avrete a disposizione strumenti molto più flessibili e pratici da usare, ma soprattutto validi per tutte le distribuzioni. Prendendo come esempio i comandi per la configurazione di sistema, avrete sicuramente notato come nel momento in cui tentate di eseguirne uno, vi venga richiesto l'accesso come *superuser* o *root*.
In Ubuntu e nelle sue derivate, per esempio, questo è identificato con il comando **sudo**, cui poi deve seguire l'inserimento della password. Altre distro come Fedora, invece, utilizzano **su**, che da accesso all'utente root fino a quando non si inserisce

Google Drive. Una volta eseguita la prima autenticazione, gsync provvederà a creare un token che permette di saltare il passaggio per i prossimi accessi. Se vuoi dare un'occhiata alle opzioni a tua disposizione, usa il comando:

```
gsync --help
```

Se hai esperienza di rsync, noterai come le istruzioni siano le stesse. Adesso puoi sincronizzare da e verso Google Drive utilizzando il desktop.

2 Riavvio bloccato

Ho installato Ubuntu 15.04. Quando tento di riavviare il computer, talvolta questo si blocca restituendomi solo una schermata nera. L'unico modo per venire fuori da tale situazione è scollegare il PC dalla rete elettrica. Sapete da cosa dipende questo malfunzionamento e come posso risolvere?

L'impossibilità a riavviare il sistema con conseguente comparsa di una schermata nera è un problema piuttosto diffuso con Ubuntu. Si tratta di un malfunzionamento causato da un'incompatibilità BIOS di alcuni portatili con la distro stessa. Tuttavia esiste un modo piuttosto rapido per risolvere. In primo luogo, apri il terminale e inserisci il comando

```
sudo gedit /etc/default/grub
```

Si apre quindi il file Grub in un editor di testo. Scorrilo fino a trovare la riga:

```
GRUB_CMDLINE_LINUX=""
```

quindi sostituiscila con:

```
GRUB_CMDLINE_LINUX="reboot=efi"
```

Adesso chiudi il file di Grub dopo averlo salvato e aggiorna da terminale con l'istruzione:

```
sudo update-grub
```

Prova a riavviare il computer e controllare che il problema non si ripresenti. Se così non fosse, apri nuovamente il file di Grub e sostituisci:

```
GRUB_CMDLINE_LINUX=""
```

con:

```
GRUB_CMDLINE_LINUX="reboot=bios"
```

oppure:

```
GRUB_CMDLINE_LINUX="reboot=acpi"
```

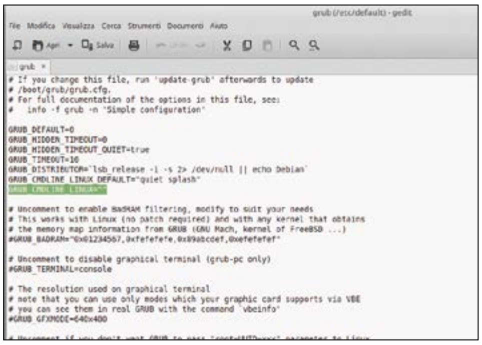
o in alternativa:

```
GRUB_CMDLINE_LINUX="reboot=pci"
```

3 Ripristinare il kernel

Ho aggiornato il kernel della mia distro Linux Mint, ma il PC si è bloccato e non ne vuole sapere di avviarsi. C'è un modo per risolvere questa spiacevole situazione senza ricorrere alla formattazione?

Certamente. Le distro basate su Ubuntu come Mint mettono a disposizione una pratica console di ripristino che consente di correre ai ripari quando si verificano situazioni di questo genere. In primo luogo, avvia il PC tenendo premuto il tasto **Maiusc**,



› **Potete agire sulle opzioni di Grub aprendo il file di configurazione in /etc/default/grub**

così da far comparire il menu di avvio di Grub da cui scegliere con quale modalità lanciare il sistema operativo. Seleziona la funzione **recovery mode**, quindi, nella schermata che si apre immediatamente dopo, scegli la voce **root**. Adesso, dal terminale inserisci il seguente comando, avendo cura di sostituire le X con la versione del Kernel che hai installato in precedenza e che impedisce l'avvio della distro.

```
mount -rw -o remount /
sudo apt-get purge linux-image-x.x.x*
```

In questo modo, avrai eliminato il kernel fallato. Nel caso non conoscessi la versione esatta del kernel che ha creato problemi, utilizza l'istruzione **uname-r**. Ora aggiorna Grub con:

```
sudo update-grub
```

Il kernel corrotto è quindi stato rimosso e non rimane che riavviare il sistema per permettere a quello precedentemente funzionante di entrare in azione.
È importante sapere che ogni qual volta si aggiorna il kernel, l'ultima versione non va a sostituirsi a quella precedente, bensì vi si affianca ponendosi solo in linea temporale successiva. Rimuovendola, il sistema non fa altro che caricare la seconda in lista. »

Riferimento rapido

Utilizzo della RAM

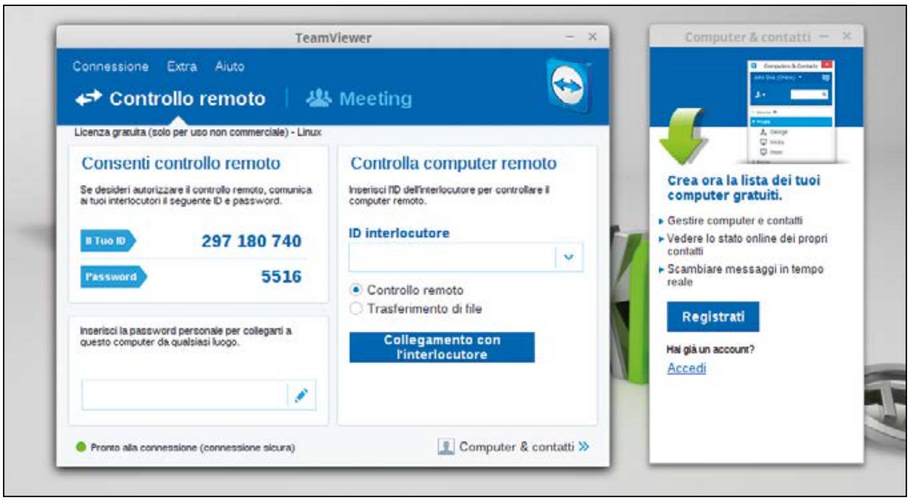
Siete preoccupati perché ogni volta che utilizzate la vostra distro e controllate l'uso della RAM rimangono solo pochi MB disponibili? Niente paura, è del tutto normale e non c'è assolutamente bisogno di acquistare nuova memoria.
È il kernel a essere responsabile di questa modalità di funzionamento. Linux odia lasciare inutilizzata la RAM, considerando questo comportamento uno spreco di risorse. Quando non ne fate uso, quindi, il kernel si occupa di allocarla ai sottosistemi così da migliorare le prestazioni. Nel momento in cui ne avrete bisogno, la RAM verrà disimpegnata e indirizzata verso i programmi che lanciate o le operazioni che eseguite. Della memoria in uso, infatti, circa la metà viene sfruttata come buffer. In tal modo, l'accesso al disco per la memorizzazione nella cache, nonché per la lettura, avviene molto più velocemente. Un altro aspetto positivo di questo modo di gestire la memoria consiste nella capacità del kernel di tenere i dati in memoria per gli utilizzi seguenti. Se avviate Firefox per due volte, vi accorgerete che al secondo tentativo il browser verrà lanciato molto più velocemente.

Domande & Risposte

4 Assistenza remota

Mi viene chiesto spesso di fornire assistenza ad alcuni amici che hanno PC Windows. Trovandosi però molto lontani da me, ho necessità di un sistema che mi permetta un accesso remoto compatibile anche con il sistema Microsoft. Sapreste darmi qualche suggerimento?

Si potrebbe pensare a VNC, ma il suo utilizzo non è dei più semplici. Infatti, in prima battuta sarebbe necessario aprire le porte necessarie, nonché lanciare un server VNC per Internet. Ci sarebbe poi bisogno di creare una VPN. Insomma, si tratta di un lavoro particolarmente gravoso, soprattutto se non si intende utilizzare l'accesso remoto per motivi professionali. Le alternative, comunque, non mancano, anche se si tratta soprattutto di programmi proprietari. Il più Open Source della categoria è **Google Chrome Remote Desktop**. Naturalmente richiede l'installazione di Chrome, più l'uso di un'estensione e un componente server. Il programma più semplice e quindi



› ***TeamViewer*** **è un ottimo programma per l'assistenza remota. Funziona bene sia su Linux sia su Windows. L'unico aspetto negativo consiste nella sua natura non Open Source**

probabilmente adatto alle esigenze di questo caso è **TeamViewer**. Se non si vuole utilizzarlo per scopi commerciali, viene messo a disposizione gratuitamente. In alternativa, dovremo pagare il costo del programma che non è proprio a buon mercato. Tuttavia non è questo il caso. Sulla tua distro Linux, potrai installare TeamViewer direttamente dal gestore pacchetti. Per quanto riguarda i PC con Windows, basta accedere alla pagina **http://teamviewer.com** e scaricare la versione per il sistema operativo Microsoft. I tuoi amici, quindi, potranno godere fin da subito di un approccio facilitato. Al primo avvio, ti troverai di fronte una finestra che guida nella modalità di accesso automatico. Puoi

La soluzione del mese

SD capricciosa

Ho comprato una scheda MicroSDXC da 64 GB. Tuttavia non riesco a farla rilevare dalla mia distro Linux. Pensavo che tutte le schede di questo genere utilizzassero il filesystem FAT MS-DOS naturalmente supportato da Linux. Sapreste dirmi come utilizzare la SD in questione? Pensate sia danneggiata?

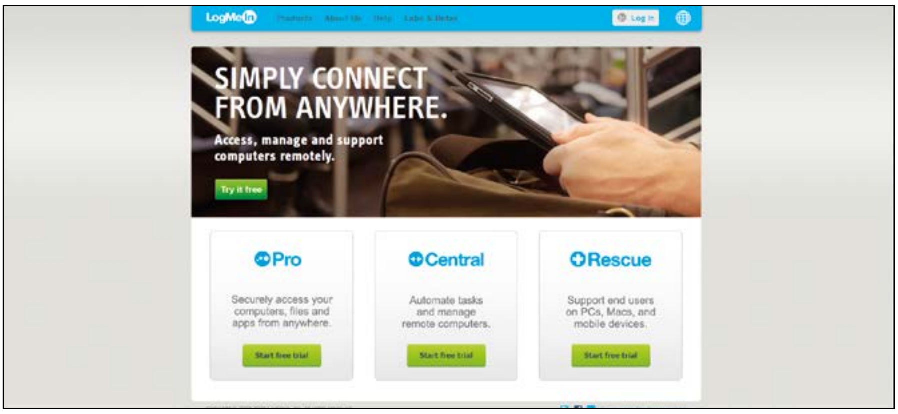
Sono gli standard SD a definire il filesystem predefinito utilizzato da queste schede. Ci sono comunque delle differenze in base alle dimensioni. SD, infatti, viene utilizzato fino a 4 GB, SDHC vale per schede che vanno da 4 a 32, mentre SDXC funziona per unità da 64 GB a 2 TB. Le schede microSD, seppure fisicamente diverse perché più piccole, seguono le stesse regole per quanto riguarda il filesystem. Per le SD e SDHC lo standard è FAT. Tuttavia è importante ricordare come quest'ultimo soffra di alcune limitazioni, una delle quali è appunto la capacità di gestire documenti non superiori ai 4 GB. Ecco quindi il motivo per cui la specifica SDXC impone l'uso di exFAT che si dimostra molto più adatto a file di grandi dimensioni. Tuttavia ha un difetto: è di proprietà e concesso in licenza da Microsoft. Per inciso, è questo il motivo per cui spesso i dispositivi hanno limiti di memoria pari a 32 GB. Pur riuscendo a supportare quantitativi superiori, non dispongono delle opportune licenze rilasciate da Redmond per superare quella specifica dimensione. Per quanto riguarda il problema posto, ci sono due possibili soluzioni. La prima consiste nel riformattare la scheda con un filesystem FAT32. È sì vero che sarai limitato a non usare file dalle dimensioni superiori ai 4 GB, ma in questo modo aumenterai a dismisura la compatibilità dell'unità, riuscendo a leggerla su qualsiasi dispositivo. L'altra opzione consiste nell'installare un driver exFAT sulla macchina Linux. Samsung, per esempio, ha pubblicato il codice sorgente GPL per un driver del kernel exFAT, ma a causa di alcune questioni riguardanti i brevetti, questo non è stato incluso nel kernel Linux. Ciò nonostante, la comunità non si è certo fermata davanti a questo problema e grazie all'uso di FUSE è riuscita a risolvere. Infatti, è sufficiente installare due pacchetti, chiamati rispettivamente **exfat-fuse** ed **exfat-utils**. Il primo è il driver FUSE, mentre il secondo contiene i programmi userspace e le varianti ExFAT di mkfs, fsck e mount. Una volta compiuta questa operazione, dovresti essere in grado di far riconoscere senza problemi la tua MicroSDXC al computer. Se non dovesse funzionare, potrebbe essere necessario caricare manualmente il modulo FUSE. Per farlo, utilizza il comando:

```
sudo modprobe fuse
```

In tal modo, il filesystem viene forzato a riconoscere il modulo in questione. Adesso devi quindi assicurarti che il componente venga avviato automaticamente quando il sistema operativo si attiva. Crea il file **/etc/modules-load.d/fuse.conf** come root e contenente "fuse" su una singola riga. Puoi seguire questo esempio:

```
echo fuse | sudo tee -a /etc/modules-load.d/fuse.conf
```

Fai però attenzione, perché non tutte le distro usano questa directory. Infatti, talvolta è possibile che assuma il nome **/etc/modules**.



› ***LogMeIn*** **è un'alternativa a TeamViewer. Peccato però sia a pagamento**

però saltarla, passando direttamente alla fase successiva. La finestra principale, infatti, mostra un ID e una password che devono essere comunicati all'utente con cui vuoi collegarti. Per evitare di scambiare i dati di accesso ogni volta, è sufficiente creare un account TeamViewer e registrare le credenziali del computer remoto cui ti connetti. Lo avrai così sempre a disposizione in una lista di PC associati. Rispetto al client per Windows, quello per Linux ha un aspetto un po' strano. Questa applicazione è infatti stata sviluppata principalmente per il sistema Microsoft. Il porting per Linux è quindi avvenuto secondariamente e senza mettere più di tanto mano all'interfaccia grafica. Ciò nonostante, TeamViewer è considerato un ottimo programma multi-piattaforma che mette a disposizione anche una versione per Android. Infatti, è possibile gestire da remoto i PC anche con tablet e smartphone che montano il sistema operativo di Google. Si tratta comunque di una procedura a senso unico. Infatti, non è possibile fare il contrario e quindi controllare il dispositivo mobile da computer. L'unico aspetto da tenere in considerazione è la natura stessa dell'offerta gratuita. Come succede spesso in questi casi, le aziende che forniscono anche versioni commerciali dei propri prodotti possono decidere di dismettere la release gratuita, lasciando a bocca asciutta tutti gli utenti che fino a quel momento ne hanno beneficiato. Un esempio calzante può essere quello di LogMeIn che ha appunto chiuso il client gratuito. Il nostro consiglio, quindi, è sempre tenersi pronto un piano B per soppiantare il programma principale. LXP

Soluzione rapida

La quantità di distro a disposizione degli amanti del mondo Linux è davvero enorme. Non passa giorno in cui non vengano rilasciate nuove versioni. Per chi si avvicina per la prima volta al Pinguino, può non essere facile scegliere quella più adatta per le proprie competenze e per l'uso che se ne intende fare. Un primo consiglio che diamo, è dare un'occhiata a **Distrowatch**, **www.distrowatch.com**. In seconda battuta, potete fare un semplice test che vi dirà quale distro è più adatta a voi. Lo trovate alla pagina **www.linux.it/scegli**. In linea generale, comunque, le migliori distribuzioni per i principianti sono Ubuntu e Linux Mint.





FAQ: le domande più frequenti

Partizioni e filesystem

› Perché su Linux dobbiamo avere tante partizioni e filesystem differenti?
Un hard disk è suddiviso in partizioni, vale a dire sezioni distinte che agiscono come se ciascuna fosse un singolo disco. Ogni partizione usa poi un filesystem per memorizzare i propri dati.

› Perché Linux utilizza più partizioni?
Mantenere i dati di swap su un filesystem dedicato è più sicuro ed efficiente rispetto a usare un file interno a una directory presente nella partizione di root.

› Quindi, alla fine, basta usare una partizione di root e una per lo swap?
In linea teorica è sufficiente, ma non consigliabile. Una buona configurazione di sistema dovrebbe almeno utilizzare una partizione diversa per la directory home. In questo modo, si mantengono le impostazioni personali separate dai dati di sistema.

› Perché dovrei farlo?
Se in un dato momento vuoi provare una distro diversa, oppure installare una versione successiva della distribuzione che utilizzi, avrai bisogno di una partizione dedicata. Questo a meno di non voler formattare il disco. Con una partizione home separata e un installer ben configurato, potrai agire senza alcun problema.

› Ci sono altre partizioni che dovrei utilizzare?
In un server, utilizzare una partizione separata per **/var** è sempre una buona soluzione. Si tratta di una procedura utile anche per i desktop, così da evitare che un processo pirata riempia senza criterio i file di log del sistema conservati in **/var/log**.

› Perché ci sono così tanti tipi di filesystem?
Ogni filesystem ha i propri punti forti e deboli. **Ext2**, per esempio, è il filesystem originale di Linux e può vantare un'eccellente velocità. Tuttavia ha il difetto di non gestire in modo ottimale gli eventuali spegnimenti improvvisi come quelli dovuti alla mancanza di corrente. **Ext3**, invece, è un filesystem *journaling*. Questo significa che, in caso di blocco, permette il recupero delle funzioni. Inoltre si dimostra più flessibile e affidabile rispetto a molte altre soluzioni. Per contro, però, ha prestazioni meno incisive.

L’eco dei LUG

I Lug

I LUG rappresentano da sempre il punto di riferimento per chiunque voglia conoscere GNU/Linux. Ogni mese dedicheremo loro questo spazio per la comunicazione di nuovi progetti e appuntamenti. Se hai qualcosa da segnalarci scrivi a **ecodeilug@linuxpro.it**

ABRUZZO
AnxaLUG - Lanciano
www.anxalug.org
Il Pinguino - Teramo
Non disponibile
MarsicaLUG - Marsica
www.marsicalug.it
OpenLUG - L’Aquila
Non disponibile
Pescara LUG
www.pescaralug.org
Pineto LUG
www.pinetolug.org
Pollinux LUG - Pollutri
Non disponibile
SSVLUG - San Salvo, Vasto, Termoli
www.ssvlug.org
SulmonaLUG
http://sulmonalug.it
TeateLUG - Chieti
Non disponibile
TeLUG - Teramo
www.telug.it
User Group Valle Roveto
http://linuxvalley-os4.blogspot.com/

BASILICATA
Basilicata LUG - Potenza e Matera
www.baslug.org

CALABRIA
3BYLug - Trebisacce
www.3bylug.tk
Bogomips - Bisignano
www.blug.it
CastroLUG
http://castrolug.altervista.org
Cosenza Hack Laboratory
http://hacklab.cosenzainrete.it/
CSLUG - Cosenza
http://cslug.linux.it
CzLug
Non disponibile
HackLab Catanzaro
http://hacklab.cz
Piana LUG - Piana di Gioia Tauro
Non disponibile
Reggio Calabria LUG
http://rclug.linux.it
Revolutionary Mind
www.revolutionarymind.org
SpixLug - Spezzano Albanese
Non disponibile

CAMPANIA
AFR@Linux LUG
www.afralinux.netsons.org
Afralug - Afragola
www.afralug.com
CasertaLUG
www.casertaglug.org
Hackaserta 81100
www.81100.eu.org
HackMeetNaples Napoli HackLab
www1.autistici.org/hmn
IGLUG - Napoli e provincia
www.iglug.org
IRLUG - Irpinia
www.irlug.it
LUG-Ischia
www.lug-ischia.org
NALUG - Napoli
www.nalug.net
Neapolis Hacklab
www.officina99.org/hacklab.html
Padulug - Paduli (BN)
http://linux.paduli.com
SCALUG - Scafati (SA)
http://xoomer.alice.it/scalug/
Tuxway.org - Provincia di Napoli
www.tuxway.org
VaLug - Vallo Linux User Group
www.valug.it
XALUG - Salerno
http://xalug.tuxlab.org

EMILIA ROMAGNA
ALFLUG - Alfonsine
www.alflug.it
Borgotaro LUG - Val Taro
http://btlug.it/
ConoscereLinux - Modena
www.conoscerelinux.it
ERLUG
http://erlug.linux.it
Ferrara LUG
www.ferrara.linux.it
FoLUG - Forlì
http://folug.linux.it
ImoLUG - Imola
www.imolug.org
LUGPiacenza
www.lugpiacenza.org
PANLUG - Vignola
Non disponibile
PLUG - Parma
http://parma.linux.it
RavennaLUG
www.ravennalug.org
RELug - Reggio Emilia e provincia
http://relug.linux.it
RiminiLug
www.riminilug.it
S.P.R.I.Te
http://sprite.csr.unibo.it
UIELinux - Valle del Rubicone
www.uielinux.org

FRIULI VENEZIA GIULIA
GOLUG - Gorizia
www.golug.it
IGLU - Udine
http://iglu.cc.uniud.it
LUG Pordenone
www.pnlug.it
LugTrieste
http://trieste.linux.it
LUG [A] [L] [P] - Aquileia
www.alproject.org

LAZIO
CiLUG - Frosinone
www.cilug.org
CLUG - Cassino
http://cassino.linux.it/
GioveLUG - Terracina
www.giovelug.org
La Sapienza LUG
www.lslug.org
Latina LUG
www.llg.it
LUG Privernum Volsca - Priverno (LT)
www.pvlug.org
LUGRoma
www.lugroma.org
LUGRoma 3
www.lugroma3.org
TorLUG - Università Tor Vergata - Roma
http://lug.uniroma2.it/
V.I.S.C.O.S.A. - Ciampino
www.viscosa.org

LIGURIA
Genuense Lug - Genova e d’intorni
http://genova.linux.it
GinLug - Genova Sampierdarena
www.sennaweb.org
Govonis GNU/LUG - Provincia di Savona
www.govonis.org
SLIMP - Software Libero Imperia
http://slimp.it/
TLug-TSL - Tigullio Ligure
http://tlug.linux.it/

LOMBARDIA
BGLug - Bergamo e provincia
www.bglug.it
BGLug Valle Seriana - Valle Seriana
http://bglugvs.web3king.com/
BrigX - Monza e Brianza
http://brigx.it
GL-Como - Como
www.gl-como.it
GLUX - Lecco e provincia
www.lecco.linux.it
GULLP - Gruppo Utenti Linux Lonate Pozzolo
www.gullp.it
IspraLUG - Ispra
http://ispralug.eu/
LIFO - Varese
www.lifolab.org
LIFOS - Cinisello Balsamo
www.lifos.org
Linux Var - Varese
www.linuxvar.it
LoLug - Lodi e provincia
www.lolug.org
Lug Bocconi - Milano
www.lug-bocconi.org
LugBS - Brescia e provincia
http://lugbs.linux.it/
Lug Castegnato - Castegnato
www.kenparker.eu/LugCastegnato
LugCR - Cremona e provincia
www.lugcr.it
Lug Crema - Crema

L'eco dei LUG

http://filibusta.crema.unimi.it/
LUGDucale - Vigevano
www.lugducale.it
LugMan - Mantova e provincia
www.lugman.org
LugOB - Cologne e ovest bresciano
www.lugob.org
MoBLUG - Monza e Brianza
www.bubblesfactory.it
OpenLabs - Milano
www.openlabs.it
POuL - Milano
www.poul.org
TiLug - Pavia
http://pavia.linux.it
VigLug - Vignate, Milano Est - Adda Martesana
www.viglug.org

MARCHE

Ascolinux LUG/FSUG Ascoli
http://marche.linux.it/ascoli/
CameLUG - Camerino
www.camelug.it
CMlug
www.cmlug.org
Egloo
www.egloo.org
FanoLUG
www.fanolug.org
Fermo LUG
www.linuxfm.org/fermolug/
GLM - Macerata
www.gruppolinuxmc.it/start/index.php
LUG Ancona
www.egloo.org
LUG Marche
http://marche.linux.it
PDP Free Software User Group
http://pdp.linux.it
SenaLug - Senigallia
www.lug.senigallia.biz

MOLISE

Campobasso LUG
http://cb.linux.it/
FrenterLUG - Larino
Non disponibile
SmaLUG - San Martino
www.smalug.org

PIEMONTE

ABC Lug - Alba/Bra/Carmagnola
http://abc.linux.it/
AlLug - Alessandria e provincia
www.allug.it
BiLUG - Provincia di Biella
http://www.bilug.it
FASoLi - Alessandria e provincia
http://softwarelibero.al.it/
Gallug - Galliate
www.gallug.it
GlugTO - Torino e provincia
www.torino.linux.it
IvLug - Ivrea Linux User Group
www.ivlug.it
Linox Novara
www.linoxnovara.org
SLIP - Pinerolo
http://pinerolo.linux.it/
ValSusinux - Val Susa e Val Sangone
www.valsusinux.it

PUGLIA

BriLUG - Brindisi
www.brilug.it
CapitanLUG - Capitanata
www.capitanlug.it
LATLUG - Latiano Linux User Group
www.latlug.org
LUGargano
www.lugargano.it
LUGBari - Bari e provincia
www.lugbari.org
MurgiaLug - Santeramo in Colle
www.open-pc.eu/index.php/murgialug/
SaLUG! - Salento
http://salug.it
Talug - Taranto
www.talug.it

SARDEGNA

GNUraghe - Oristano
www.gnuraghe.org
GULCh - Cagliari
www.gulch.it
PLUGS - Sassari
www.plugs.it
GULMh - Macomer (NU)
www.gulmh.org

SICILIA

CefaLug - Cefalù
http://cefalug.linux.it
cLUG - Caltanissetta
www.clug.it
EnnaLUG
www.ennalug.org
FreakNet MediaLab - Catania
www.freaknet.org
Leonforte LUG
http://leonforte.linux.it
LUG Catania
www.catania.linux.it
LUGSR - Siracusa
www.siracusa.linux.it
MELUG - Messina
Non disponibile
Norp LUG - Noto, Pachino, Rosolini
Non disponibile
PALUG - Palermo
http://palermo.linux.it
RgLUG - Ragusa e provincia
http://ragusa.linux.it
VPLUG Linux Planet - Provincia Caltanisetta
www.vplug.it
SputniX - Palermo
www.sputnix.it

TOSCANA

ACROS - Versilia, Lucca, Massa Carrara
www.lug-acros.org
Elbalinux
Non disponibile
ElsaGLUG - Val d'Elsa
www.elsaglug.org
FLUG - Firenze
www.firenze.linux.it
GOLEM - Empoli, Valdelsa
http://golem.linux.it
GroLUG - Grosseto
www.grolug.org
G.U.L.LI - Livorno
www.livorno.linux.it
GulP! Piombino
http://gulp.perlmonk.org
GULP Pisa
www.gulp.linux.it
GuruAtWork - Grosseto e provincia
www.guruatwork.com
IPIOS - Bibbiena e valle del Casentino
www.ipios.org
Lucca LUG
http://luccalug.it
L.U.G.A.R - Arezzo
Non disponibile
PLUG - Prato e provincia
www.prato.linux.it
PtLug - Pistoia e provincia
www.ptlug.org
SLUG - Siena e provincia
www.siena.linux.it

TRENTINO ALTO ADIGE

LinuxTrent - Trento
http://linuxtrent.it
LugBz - Bolzano
www.lugbz.org

UMBRIA

OrvietoLUG
www.orvietolug.it
LUG Perugia
www.perugiagnulug.org
TerniLUG
www.ternignulug.org

VALLE D'AOSTA

SLAG - Aosta
www.slag.it

VENETO

0421ug - Provincia di Venezia
www.0421ug.org
BLUG - Belluno
http://belluno.linux.it
Faber Libertatis - Padova
http://faberlibertatis.org
GrappaLUG - Bassano del Grappa
http://grappalug.homelinux.net/
ILC - Informatica Libera Cittadellese - FSUG
http://ilc.pd.it
LegnagoLUG
Non disponibile
Linux Ludus - Villafranca (VR)
www.linuxludus.it
LugAnegA
www.luganega.org
LUGSF - San Fidenzio
Non disponibile
LUG Vicenza
www.vicenza.linux.it
LugVR - Verona
www.verona.linux.it
MontelLUG - Montebelluna
www.montellug.it
FSUG Padova
www.fsugpadova.org
RoLUG - Rovigo
http://rovigo.linux.it
TVLUG - Treviso
www.tvlug.it
VELug - Venezia
www.velug.it
AViLUG Schio
http://www.avilug.it/doku.php

NAZIONALI

FSUGitalia
www.fsugitalia.org
Gentoo Channel Italia
www.gechi.it
MajaGLUG
www.majaglug.net
SkyLUG
http://tech.groups.yahoo.com/group/skylug/

Guida DVD

Ogni mese Linux Pro vi offre i programmi e le distribuzioni più recenti su DVD

Ogni volta che troverete questo simbolo in un articolo, vorrà dire che i file citati si trovano nel DVD allegato alla rivista.

Distro Desktop

Korora 22

Korora 22, nome in codice Selina, è stata rilasciata ufficialmente il 2 agosto. Si tratta di una distribuzione *user-friendly* basata su Fedora. Il suo punto forte è la disponibilità di una vasta serie di desktop, tra cui: GNOME, KDE5, MATE e XFce.

Le principali novità

Una volta scaricata la distro con il desktop che si preferisce, potete utilizzarla sia in modalità Live sia procedere immediatamente con l'installazione su disco. Quest'ultima è in puro stile Fedora, di cui peraltro riprende l'installer. Non avrete quindi alcuna difficoltà a portarla a termine con tanto di partizionamento manuale. Rispetto alla versione precedente, Korora 22 non supporta più KDE4, ma passa direttamente alla versione Plasma 5. Per quanto riguarda gli altri desktop, abbiamo la versione 2.6 di Cinnamon, la 1.10 di MATE, la 3.16 di GNOME e Xfce 4.12. Gli sviluppatori hanno poi dichiarato di aver rimosso il plug-in Flash. Il motivo di questa scelta è quanto mai giustificato. Infatti, oramai non serve più a niente se non a veicolare virus e affini. Ciò nonostante, per tutti coloro che non ne vogliono fare a meno, il repository che lo contiene è abilitato e quindi, per installarlo, basta agire dal gestore pacchetti. In alternativa, eseguite il comando **sudo dnf install flash-plugin**. Come si può capire dall'analisi di questa istruzione, Korora 22 introduce l'uso di DNF e Hawkey. DNF è molto simile al gestore pacchetti Yum ma è stato riscritto per fornire prestazioni ottimali. Infatti, fornisce una definizione estremamente rigorosa per API, plug-in e progetti di vario genere. DNF, inoltre, sfrutta la libreria **libsolv** che è stata inizialmente introdotta dal progetto OpenSUSE per fornire una gestione delle dipendenze più veloce e flessibile. Rispetto a Yum, inoltre, DNF vanta anche una migliore gestione ed è progettato per usare un codice più pulito e pratico anche per i principianti. **Elastisearch** è un'altra caratteristica molto interessante di Korora 22. Si tratta di un popolare server di indicizzazione Open Source che in precedenza era disponibile solo per Yum. Adesso, per installarlo, è invece possibile utilizzare il comando **dnf install elasticsearch**.

Firewall a prova di bomba

Il firewall integrato in Korora 22 è di alto profilo. In primo luogo è rivolto agli sviluppatori ed è configurato sia per mantenere la massima sicurezza, sia per permettere svariati test sulle porte. Non solo, ma è progettato per permettere l'interconnettività flessibile con i più moderni dispositivi. Le porte sotto la 1024, con l'eccezione di sshd, Samba e DHCPv6, sono bloccate per evitare l'accesso da parte di terzi ai servizi di sistema. Quelle superiori alla 1024, invece, vengono utilizzate per le applicazioni avviate dall'utente e sono aperte per impostazione predefinita. Per personalizzare il funzionamento del firewall, è poi disponibile una nutrita documentazione che potete trovare alla pagina **https://fedoraproject.org/wiki/FirewallD**. Se invece volete utilizzare lo strumento grafico di configurazione, basta sfruttare il pacchetto **firewall-config**.

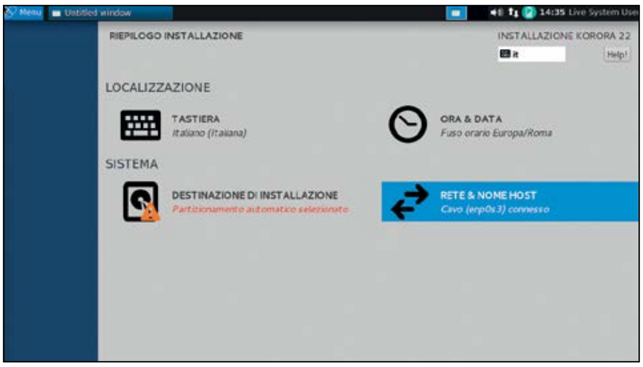
Software per tutti

A livello di applicazioni, ci troviamo di fronte a una distro ricca di software inclusi. Per quanto riguarda il comparto Internet troviamo Ekiga, Firefox, Gwibber, Liferea, Pidgin, Thunderbird e Transmission. Non manca poi la suite completa di LibreOffice, più Dictionary e FBReader per la lettura degli eBook. Trovate poi il semplice editor di testo Mousepad, nonché Xfburn per la masterizzazione di CD e DVD. La documentazione, infine, è disponibile nella sezione **Support** del sito.

Cosa c'è nel DVD

- Audacity 2.1.0
- Bracket 1.2
- Chromixium 1.5
- Empathy 3.12.3
- Fiziko 0.1
- FreeCol 0.11.5
- Freemind 1.0.1
- Geary 0.10
- Jitsi 2.9
- Kopete 0.12.2
- Korora 22
- Linphone 3.8.5
- Netsurf 3.3
- Pasang Emas 5.0
- Pidgin 2.10.11
- Psi 0.15
- REMnux
- Shotwell 0.22.0
- VirtualBox 5.0 Beta
- Warsow 1.51

Cosa ci offre Korora 22



1 Installazione by Fedora

L'installer è lo stesso di Fedora. Il processo è quindi scandito dalle istruzioni grafiche che vi portano all'installazione della distro su hard disk. Da sottolineare la presenza dello strumento di partizionamento avanzato che consente di criptare le porzioni del disco.



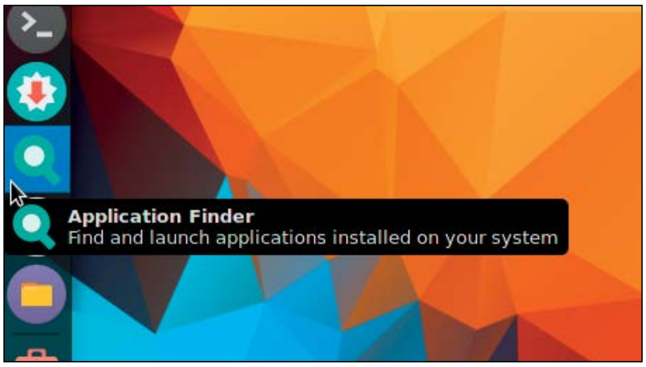
2 Ambiente colorato

Il desktop che abbiamo scelto per questa installazione è Xfce, ma ricordate che sono disponibili anche Cinnamon, Mate, GNOME e KDE5. L'ambiente è pulito e informale, con un sfondo del desktop colorato e che mette subito a proprio agio anche i principianti.



3 Menu funzionale

Il menu è disposto nella parte superiore sinistra dell'interfaccia. Mantiene lo stesso approccio semplificato e colorato del desktop. Le applicazioni sono ordinate per sezioni e vengono immediatamente identificate dalle icone personalizzate che gli sviluppatori hanno introdotto.



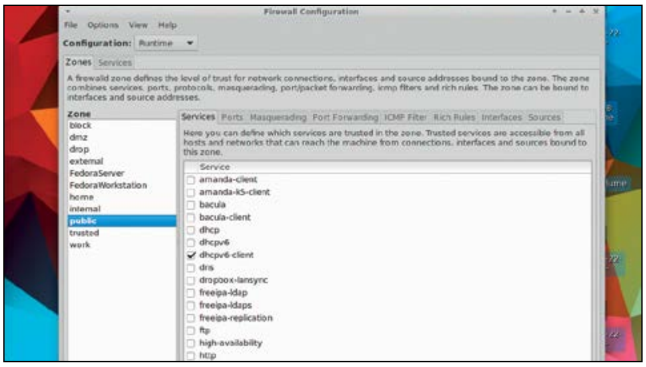
4 Launcher a comparsa

Se collocate il puntatore del mouse sul bordo sinistro, si apre un comodo Launcher che consente di accedere alle applicazioni più utili. Rimane in sovraimpressione anche quando sono in esecuzione altre applicazioni e può essere sfruttato in modo rapido e veloce per accedere al Finder.



5 Impostazioni sempre sotto mano

Il pannello delle impostazioni non ha una propria finestra, ma consente effettivamente di accedere alle varie funzioni direttamente dal menu principale. Sulla sinistra, infatti, sono disponibili anche le voci per personalizzare plug-in specifici come **IcedTea** o **OpenJDK8**.



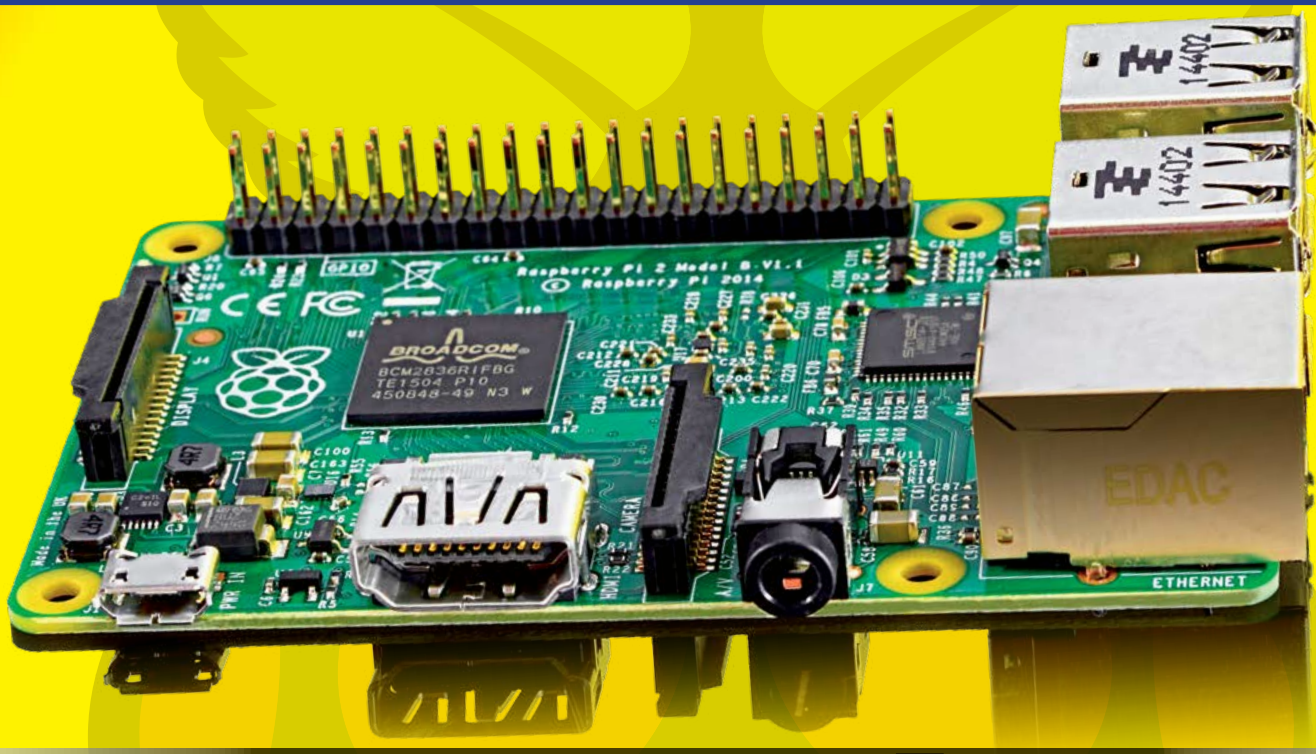
6 Firewall configurabile

Il pacchetto **firewall-config** per la gestione grafica del firewall di Korora 22 è già integrato nel sistema e accessibile dal pannello delle Impostazioni. Utilizzandolo, potete mettere mano al comportamento di questo componente e piegarlo totalmente ai vostri voleri. LXP

In edicola il 13 ottobre

NEL PROSSIMO NUMERO

Hai un argomento da proporci?
Scrivi a prossimamente@linux.it



I PROGETTI PIÙ BELLI PER RASPBERRY PI

Mettiamo il computer più piccolo e potente del mondo a buon uso realizzando progetti semplici e complessi per ogni tipo di necessità!

LINUX PRO

Mensile, prezzo di copertina 5,90 €
www.linuxpro.it

Direttore responsabile: *Luca Sprea*

Traduzione e Localizzazionea cura di:
Ventidodici di Andrea Orchesi
redazione@linuxpro.it

Sprea S.p.A.
Socio Unico - direzione e coordinamento di Sprea Holding S.p.A.

Presidente: Luca Sprea
Consigliere delegato: Mario Sprea

Coordinamento:
Gabriella Re (Foreign Rights) international@sprea.it, Alberta Rivolta (PA), Ambra Palermi (Segreteria Editoriale), Francesca Sigismondi (Ufficio Legale), Tiziana Rosato (acquisti e produzione), Emanuela Mapelli (Pianificazione Pubblicitaria)

Amministrazione: Anna Nese (CFO), Erika Colombo (controller), Irene Citino, Sara Palestra
amministrazione@sprea.it

Servizio qualità edicolanti e DL: Sonia Lancellotti, Andrea Palermi
distribuzione@sprea.it

PUBBLICITA' E MARKETING:
Alessandra Cappellacci- alessandracappellacci@sprea.it
Tel.0292432275 - Cell.3204670523

Sede Legale: - via Torino, 51 20063 Cernusco Sul Naviglio (Mi) - Italia
PI 12770820152- Iscrizione camera Commercio 00746350149

ABBONAMENTI E ARRETRATI
Servizio qualità abbonamenti e arretrati: Desirée Conti

Abbonamenti: si sottoscrivono on-line su www.linuxpro.it/abbonamenti
Mail: abbonamenti@linuxpro.it
Fax: 02 700 53 76 72
Tel: 02 87 15 82 25 (lun-ven / 9:00-18:00)

Arretrati: si acquistano on-line su www.linuxpro.it/arretrati
Mail: arretrati@linuxpro.it
Fax: 02 700 53 76 72 - Tel: 02 87 15 82 25 (lun-ven / 9:00-18:00)
Il prezzo dell'abbonamento è calcolato in modo etico perché sia un servizio utile e non in concorrenza con la distribuzione in edicola.

www.myabb.it a cura di Aktia srl
Contenuti su licenza: "Linux Format"- Future Publishing Limited PLC., Bath UK

Registrazione testata: Linux Pro, pubblicazione mensile registrata al Tribunale di Milano il 08.02.2003 con il numero 74.

Distributore per l'Italia e per l'estero:
Press-Di Distribuzione stampa e multimedia s.r.l. - 20134 Milano
ISSN: 1722-6163

Stampa: Arti Grafiche Boccia S.p.A.- Salerno